

Frontiers
in
Artificial
Intelligence
and
Applications

INFORMATION MODELLING AND KNOWLEDGE BASES XX

Edited by
Yasushi Kiyoki
Takahiro Tokuda
Hannu Jaakkola
Xing Chen
Naofumi Yoshida

IOS
Press

VISIT...

LANZAROTE
Caliente.COM

INFORMATION MODELLING AND KNOWLEDGE
BASES XX

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including “Information Modelling and Knowledge Bases” and “Knowledge-Based Intelligent Engineering Systems”. It also includes the biennial ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Series Editors:

J. Breuker, R. Dieng-Kuntz, N. Guarino, J.N. Kok, J. Liu, R. López de Mántaras,
R. Mizoguchi, M. Musen, S.K. Pal and N. Zhong

Volume 190

Recently published in this series

- Vol. 189. E. Francesconi et al. (Eds.), Legal Knowledge and Information Systems – JURIX 2008: The Twenty-First Annual Conference
- Vol. 188. J. Breuker et al. (Eds.), Law, Ontologies and the Semantic Web – Channelling the Legal Information Flood
- Vol. 187. H.-M. Haav and A. Kalja (Eds.), Databases and Information Systems V – Selected Papers from the Eighth International Baltic Conference, DB&IS 2008
- Vol. 186. G. Lambert-Torres et al. (Eds.), Advances in Technological Applications of Logical and Intelligent Systems – Selected Papers from the Sixth Congress on Logic Applied to Technology
- Vol. 185. A. Biere et al. (Eds.), Handbook of Satisfiability
- Vol. 184. T. Alsinet, J. Puyol-Gruart and C. Torras (Eds.), Artificial Intelligence Research and Development – Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence
- Vol. 183. C. Eschenbach and M. Grüninger (Eds.), Formal Ontology in Information Systems – Proceedings of the Fifth International Conference (FOIS 2008)
- Vol. 182. H. Fujita and I. Zulkernan (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the seventh SoMeT_08
- Vol. 181. A. Zgrzywa, K. Choroś and A. Siemiński (Eds.), New Trends in Multimedia and Network Information Systems
- Vol. 180. M. Virvou and T. Nakamura (Eds.), Knowledge-Based Software Engineering – Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering

ISSN 0922-6389

Information Modelling and Knowledge Bases XX

Edited by

Yasushi Kiyoki

Keio University, Japan

Takahiro Tokuda

Tokyo Institute of Technology, Japan

Hannu Jaakkola

Tampere University of Technology, Finland

Xing Chen

Kanagawa Institute of Technology, Japan

and

Naofumi Yoshida

Komazawa University, Japan

IOS
Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

© 2009 The authors and IOS Press.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without prior written permission from the publisher.

ISBN 978-1-58603-957-8

Library of Congress Control Number: 2008941920

Publisher

IOS Press

Nieuwe Hemweg 6B

1013 BG Amsterdam

Netherlands

fax: +31 20 687 0019

e-mail: order@iospress.nl

Distributor in the UK and Ireland

Gazelle Books Services Ltd.

White Cross Mills

Hightown

Lancaster LA1 4XS

United Kingdom

fax: +44 1524 63232

e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada

IOS Press, Inc.

4502 Rachael Manor Drive

Fairfax, VA 22032

USA

fax: +1 703 323 3668

e-mail: iosbooks@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

In the last decades information modelling and knowledge bases have become essentially important subjects not only in academic communities related to information systems and computer science but also in the business area where information technology is applied.

The 18th European-Japanese Conference on Information Modelling and Knowledge Bases (EJC 2008) continues the series of events that originally started as a co-operation initiative between Japan and Finland, already in the last half of the 1980's. Later (1991) the geographical scope of these conferences expanded to cover the whole Europe and other countries as well.

The EJC conferences constitute a world-wide research forum for the exchange of scientific results and experiences achieved in computer science and other related disciplines using innovative methods and progressive approaches. In this way a platform has been established drawing together researches as well as practitioners dealing with information modelling and knowledge bases. The main topics of EJC conferences target the variety of themes in the domain of information modelling, conceptual analysis, multimedia knowledge bases, design and specification of information systems, multimedia information modelling, multimedia systems, ontology, software engineering, knowledge and process management. We also aim at applying new progressive theories. To this end much attention is paid also to theoretical disciplines including cognitive science, artificial intelligence, logic, linguistics and analytical philosophy.

In order to achieve EJC targets, an international program committee selected 17 full papers, 6 short papers, 7 position papers in a rigorous reviewing process from 37 submissions. The selected papers cover many areas of information modelling, namely theory of concepts, database semantics, knowledge representation, software engineering, WWW information management, multimedia information retrieval, ontological technology, image databases, temporal and spatial databases, document data management, process management, and many others.

The conference could not be a success without the effort of many people and organizations.

In the Program Committee, 31 reputable researchers devoted a good deal of effort to the review process selecting the best papers and creating the EJC 2008 program. We are very grateful to them. Professor Yasushi Kiyoki, professor Takehiro Tokuda and professor Hannu Kangassalo were acting as co-chairs of the program committee. Dr. Naofumi Yoshida and his team in Program Coordination Team were managing the review process and the conference program. Professor Xing Chen was managing the conference venue and arrangement in the organizing committee. Professor Hannu Jakkola acted as a general organizing chair and Ms. Ulla Nevanranta as a conference secretary for general organizational things necessary for annually running the conference series. We gratefully appreciate the efforts of all the supporters.

We believe that the conference will be productive and fruitful in the advance of research and application of information modelling and knowledge bases.

The Editors
Yasushi Kiyoki
Takahiro Tokuda
Hannu Jaakkola
Xing Chen
Naofumi Yoshida

Organization

General Program Chair

Hannu Kangassalo, University of Tampere, Finland

Co-Chairs

Yasushi Kiyoki, Keio University, Japan

Takahiro Tokuda, Tokyo Institute of Technology, Japan

Program Committee

Maria Bielikova, Slovak University of Technology, Slovakia

Boštjan Brumen, University of Maribor, Slovenia

Pierre-Jean Charrel, University of Toulouse and IRIT, France

Xing Chen, Kanagawa Institute of Technology, Japan

Daniela Ďuráková, VSB – Technical University Ostrava, Czech Republic

Marie Duží, VSB – Technical University of Ostrava, Czech Republic

Hele-Mai Haav, Institute of Cybernetics at Tallinn University of Technology, Estonia

Roland Hausser, Erlangen University, Germany

Anneli Heimbürger, University of Jyväskylä, Finland

Jaak Henno, Tallinn University of Technology, Estonia

Yoshihide Hosokawa, Nagoya Institute of Technology, Japan

Hannu Jaakkola, Tampere University of Technology, Pori, Finland

Ahto Kalja, Tallinn University of Technology, Estonia

Hannu Kangassalo, University of Tampere, Finland

Eiji Kawaguchi, Kyushu Institute of Technology, Japan

Mauri Leppänen, University of Jyväskylä, Finland

Sebastian Link, Massey University, New Zealand

Tommi Mikkonen, Tampere University of Technology, Finland

Jørgen Fischer Nilsson, Denmark Technical University, Denmark

Jari Palomäki, Tampere University of Technology, Pori, Finland

Ita Richardsson, University of Limerick, Ireland

Hideyasu Sasaki, Ritsumeikan University, Japan

Tetsuya Suzuki, Shibaura Institute of Technology, Japan

Bernhard Thalheim, Christian-Albrechts University Kiel, Germany

Pasi Tyrväinen, University of Jyväskylä, Finland

Peter Vojtas, Charles University Prague, Czech Republic

Benkt Wangler, Skövde University, Sweden

Yoshimichi Watanabe, University of Yamanashi, Japan

Naofumi Yoshida, Komazawa University, Japan

General Organizing Chair

Hannu Jaakkola, Tampere University of Technology, Pori, Finland

Organizing Committee

Xing Chen, Kanagawa Institute of Technology, Japan
Ulla Nevanranta, Tampere University of Technology, Pori, Finland

Program Coordination Team

Naofumi Yoshida, Komazawa University, Japan
Xing Chen, Kanagawa Institute of Technology, Japan
Anneli Heimbürger, University of Jyväskylä, Finland
Jari Palomäki, Tampere University of Technology, Pori, Finland
Teppo Räisänen, University of Oulu, Finland
Daniela Ďuráková, VSB – Technical University Ostrava, Czech Republic
Tomoya Noro, Tokyo Institute of Technology, Japan
Turkka Napila, University of Tampere, Finland
Jukka Aaltonen, University of Lapland, Finland

Steering Committee

Professor Eiji Kawaguchi, Kyushu Institute of Technology, Japan
Professor Hannu Kangassalo, University of Tampere, Finland
Professor Hannu Jaakkola, Tampere University of Technology, Pori, Finland
Professor Setsuo Ohsuga, Japan

External Reviewers

Martin Necasky
Ivan Kapustik
Johannes Handl
Tarmo Robal

Contents

Preface	v
<i>Yasushi Kiyoki, Takahiro Tokuda, Hannu Jaakkola, Xing Chen and Naofumi Yoshida</i>	
Organization	vii
Towards Semantic Wikis: Modelling Intentions, Topics, and Origin in Content Management Systems	1
<i>Gunar Fiedler and Bernhard Thalheim</i>	
Center Fragments for Upscaling and Verification in Database Semantics	22
<i>Roland Hausser</i>	
Concepts and Ontologies	45
<i>Marie Duží and Pavel Materna</i>	
Conceptual Modeling of IS-A Hierarchies for XML	65
<i>Martin Necasky and Jaroslav Pokorny</i>	
Boolean Constraints for XML Modeling	85
<i>Sven Hartmann, Sebastian Link and Thu Trinh</i>	
An Image-Query Creation Method for Representing Impression by Color-Based Combination of Multiple Images	105
<i>Shiori Sasaki, Yoshiko Itabashi, Yasushi Kiyoki and Xing Chen</i>	
Construction of Peer-to-Peer Systems for Knowledge Resource Distribution Using Overlay Clustering of Similar Peers	113
<i>Huijun Li, Samir Ibradic, Xiao Shao and Takehiro Tokuda</i>	
Co-Design of Web Information Systems Supported by SPICE	123
<i>Gunar Fiedler, Hannu Jaakkola, Timo Mäkinen, Bernhard Thalheim and Timo Varkoi</i>	
A Description Logic with Concept Instance Ordering and Top-k Restriction	139
<i>Veronika Vaneková and Peter Vojtáš</i>	
3C-Drive: New Model for Driver's Auto Evaluation	154
<i>Juliette Brezillon, Patrick Brezillon and Charles Tijss</i>	
The <i>TIL-Script</i> Language	166
<i>Nikola Ciprich, Marie Duží and Michal Košinár</i>	
An Efficient Method for Quick Construction of Web Services	180
<i>Hao Han, Yohei Kotake and Takehiro Tokuda</i>	
A News Index System for Global Comparisons of Many Major Topics on the Earth	194
<i>Tomoya Noro, Bin Liu, Yosuke Nakagawa, Hao Han and Takehiro Tokuda</i>	

Toward Automatic Expertise Identification of Blogger <i>Chia Chun Shih, Jay Stu, Wen-Tai Hsieh, Wei Shen Lai, Shih-Chun Chou and Tse-Ming Tsai</i>	212
Managing Co-Reference Knowledge for Data Integration <i>Carlo Meghini, Martin Doerr and Nicolas Spyratos</i>	224
On Reducing Relationships to Property Ascriptions <i>Jørgen Fischer Nilsson</i>	245
A Domain-Specific Knowledge Space Creation Process for Semantic Associative Search <i>Minoru Kawamoto and Yasushi Kiyoki</i>	253
The Architecture of an Intelligent Agent in MAS <i>Nikola Ciprich, Marie Duží, Tomáš Frydrych, Ondřej Kohut and Michal Košinár</i>	261
A Meta-Level Knowledge Base System for Discovering Personal Career Opportunities by Connecting and Analyzing Occupational and Educational Databases <i>Yusuke Takahashi and Yasushi Kiyoki</i>	270
Temporal Entities in the Context of Cross-Cultural Meetings and Negotiations <i>Anneli Heimbürger</i>	290
A Common Framework for Board Games and Argumentation Games <i>Jenny Eriksson Lundström, Jørgen Fischer Nilsson and Andreas Hamfelt</i>	309
The Problem of Tacit Knowledge – Is It Possible to Externalize Tacit Knowledge? <i>Ilkka Virtanen</i>	321
Information System in Security Area Using Semantic Approach <i>Ladislav Buřita and Vojtěch Ondryhal</i>	331
How Psychology and Cognition Can Inform the Creation of Ontologies in Semantic Technologies <i>Paula C. Engelbrecht and Itiel E. Dror</i>	340
Knowledge, Accountability, and Relevance Systems – Objectivations of Social Reality Through Shared Symbolic Devices <i>Hakim Hachour</i>	348
Inheritance and Polymorphism in Datalog: An Experience in Model Management <i>Paolo Atzeni and Giorgio Gianforme</i>	354
A Proposal for a User Oriented Language Based on the Lyee Theory <i>Keizo Yamada, Jun Sasaki, Michiru Tanaka and Yutaka Funyu</i>	359
Towards Information Security Ontology in Business Networks <i>Jukka Aaltonen, Oliver Krone and Pekka Mustonen</i>	366
The Mouth Articulatory Modelling and Phonosemantic Conceptualization as In-Formation of Human Language <i>Alexei Medvedev</i>	373

Information Modelling for Preference Analysis of Musical Instrument Sounds <i>Yukari Shiota</i>	379
Opera of Meaning: Film and Music Performance with Semantic Associative Search <i>Shlomo Dubnov and Yasushi Kiyoki</i>	384
Intelligence and Language – How Could Human Being Have Language? <i>Setsuo Ohsuga</i>	392
Multi-Agent Knowledge Modelling <i>Marie Duží, Anneli Heimbürger, Takehiro Tokuda, Peter Vojtáš and Naofumi Yoshida</i>	411
Information Modelling and Global Risk Management Systems <i>Hannu Jaakkola, Bernhard Thalheim, Yutaka Kidawara, Koji Zettsu, Xing Chen and Anneli Heimbürger</i>	429
Subject Index	447
Author Index	449

This page intentionally left blank

Towards Semantic Wikis: Modelling Intensions, Topics, and Origin in Content Management Systems

Gunar FIEDLER and Bernhard THALHEIM

*Department of Computer Science, Christian-Albrechts University at Kiel,
Olshausenstr. 40, 24098 Kiel, Germany
{fiedler, thalheim}@is.informatik.uni-kiel.de*

Abstract. Content management is the process of handling information within an organization or community. Therefore, content management systems have to provide generic functionality for generation, extraction, storage, and exchange of digital assets. Because of the heterogeneity and complexity of content, a sufficient semantical and user-oriented annotation of content is crucial. Although semantical annotation by metadata and ontologies together with reasoning support has been extensively studied for a long time, commercially available content management systems provide only basic support for semantic modelling. Conceptual aspects of content users and support of user specific intensions are neglected. In this paper we will analyze the mismatch between the requirements of content management and semantical description and propose a data model for content which treats semantic information not only as describing metadata but incorporates the data itself, the intension behind the data, the usage of data and the origin of data on the same level.

1. Introduction

Content Management

Content in its actual definition is any kind of information that is shared within a community or organization. In difference to data in classical database systems content usually refers to aggregated macro data which is complex structured. Structuring of content can be distinguished:

- The structure of the aggregated micro data is preserved but micro data was combined to build larger chunks of information. Examples are scientific data sets such as time series of certain measurements. There is a common (or even individual) structuring and meaning for each sampling vector but the compound of all sampling vectors adds additional semantics.
- The structure of content is only partially known. A typical example is the content of Web pages: structuring is known up to a certain level of detail which may also be varying within one instance.
- Content may be subsymbolic, such as pictures, videos, music or other multimedia content.

Aggregation of content usually takes place by combining reusable fragments provided by different sources in different formats such as texts, pictures, video streams or structured data from databases. Content is subject to a content life cycle which implies a persistent change process to the content available in a content management system (CMS).

Currently, many systems claim to be content management systems. A recent overview of the German market (www.contentmanager.de, viewed June 12th, 2007) reveals hundreds of products related to tasks of content management. Most products are related to Web content management. These products organize content for Web pages with a strong orientation on editorial components such as texts and pictures.

The more generic ones agree in a major paradigm: the separation of data management and presentation management. Data management reflects the process of supporting content creation, content structuring, content versioning, and content distribution while presentation management grabs the data for delivering it to the user in various ways. Only content which is generated following this separation can be easily shared, distributed, and reused.

Following new trends and developments in Web technologies, e.g., in the context of Web 2.0 or the Semantic Web the automated processing of content becomes more and more important. Because content represents valuable assets it may be reused in different contexts (*content syndication*) or has to remain accessible for a long time.

The semistructured or even unstructured nature of content requires annotations to enable search facilities for content. Expressing semantics in a machine interpretable way has been under investigation since the early days of artificial intelligence, see e.g., [22] for a survey of knowledge representation techniques such as logical theories, rule-based systems, frames or semantic nets. Today systems handle semantical descriptions as meta-data describing certain content instances. There are different ways for associating data and metadata:

- A conceptual, logical, or physical *schema* is defined and instances are created according to this schema. This is the usual way for classical databases. The modelling language strongly restricts the capabilities of this description facility. Common languages such as Entity-Relationship Modelling or UML focus on structural properties with support of selected integrity constraints.
- Defining a schema is not applicable (or only in a restricted way) to semistructured or unstructured content. For that reason content instances are annotated. An annotation is a triple (S, P, O) where S denotes the subject to be annotated, P a predicate denoting the role or purpose of this annotation, and O the object (or resource) which is associated with S . The vocabulary for annotations is organized in ontologies and thesauri. A typical language for expressing annotations in the context of the Semantic Web is the Resource Description Framework (RDF, [31]) while the Web Ontology Language OWL ([30]) may be used to express semantic relationships between the concepts and resources used for annotation. There exist myriads of ontologies and parameter definitions for different application domains such as the Dublin Core parameters [4]) for editorial content.

Content Management and Semantic Annotations

Semantic annotation in current content management systems is usually restricted to pre-selected ontologies and parameter sets. Rich conceptual data models are only available

in more sophisticated systems. Because most generic CMS are focused on Web content management semantic annotation is usually restricted to editorial parameters. Specialized content management systems which are adapted to certain application domains incorporate preselected and tailored ontologies. Especially for XML-based content there exist several annotation platforms which incorporate semantical annotation either manually or semi-automatically; see [16] for a survey on available platforms.

Automated processing of semantical metadata is usually restricted to search facilities, e.g., searching for the author of an article. Because ontologies are preselected for most systems a full-featured reasoning support is usually not available. Especially for OWL ontologies there are reasoning tools based on description logics such as Racer ([9]) or FaCT which enable T-box (but also A-box) reasoning about semantic relationships between annotation concepts.

Applying generic semantical annotation and classical reasoning facilities to content management suffers from several drawbacks:

- Content as aggregated macro data is only partially analysable. The purpose of metadata is the description of properties which cannot be concluded from the data itself. The very simple annotation frame of (S, P, O) triples does not allow one to express complex properties. For that reason this information has to be kept in the underlying ontology by defining appropriate concepts. The support of user-specific concepts increases the size of the ontology significantly and makes reasoning support even harder. Ad hoc definitions of user-specific concepts is not supported in this annotation model.
- Annotation with respect to arbitrary ontologies implies general purpose reasoning support by the system. Reasoning for even simple languages suffers from its high computational complexity (e.g., NEXPTIME for the restricted OWL-DL dialect, [11]). Dealing with high worst-case complexities implies a small size of input data but this is a contradiction to expressible ontologies and the definition of content as complex structured macro data. Especially the size of content instances is a crucial factor because A-box reasoning is a critical point for automated content processing ([10]).

But there are advantages, too:

- Usually, it is possible to distinguish between different points of view on content instances. Not every property is important while looking from every point of view. The macro data may encapsulate and hide properties from its aggregated micro data. Reasoning about the properties of the compound can be separated from the properties of the elements as well as the properties of interconnections between content instances.
- Typical application scenarios determine important properties and suggest evaluation strategies. So ontologies may be decomposed to enable a contextualized reasoning, e.g., on the basis of Local Model Semantics ([8]). Local reasoning may rely on a language that is just as expressive as needed in this context. Contexts relying on less expressive languages may support automated reasoning while contexts relying on more expressive languages may be used for manually interpreted information. Soundness and completeness of the reasoning process are not of primary interest as long as the reasoning result is acceptable in the application domain.

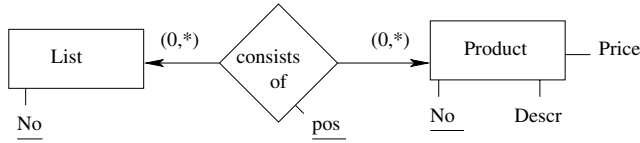


Figure 1. Schema Fragment: List of Products.

- The separation between annotations relying on common knowledge, user-specific annotations and (especially) usage-specific annotations reduces the size of incorporated ontologies significantly.
- If semantic annotations themselves are given a more sophisticated internal structure reasoning can be adapted to the requirements of the application domain.

The major disadvantage of current semantic description in content management is the treatment of knowledge over content instances as metadata on a secondary level in a strongly restricted language. In the following sections we will introduce a data model for content which handles the semantic part on the same level as the content itself and gives additional structure to the semantic description. We will start with the definition of content chunks as semantically enriched content instances in Section 2. In Section 4 we will introduce the notion of a schema for content chunks to incorporate typical functionality of content management systems such as content generation, content delivery, or content exchange.

As an example for a content management system we will take a look at a (simplified) Web shop application which sells products to customers via a website. The usual functionality should be supported: customers are able to search and browse for products, manage their profiles, shopping carts, and wish lists and order products.

2. Content Chunks

If we consider the HERM ([24]) schema fragment in Fig. 1 we see a modelled list of products.

This modelling reveals certain structural properties such as attributes of the entity types and the connection between the list and the products. But the purpose of this model is missing. What kind of list was modelled: a shopping cart, a wish list, a list of stock items? Why was it modelled? What does the modeler expect? All this information is missing in the modelled fragment but is crucial if content instances of this schema are processed: if the list represents a shopping cart, pricing information may be collected. If it represents a wish list, there may be the need for additional functionality for discovering related products. It is obvious that expressing all this information by (S, P, O) annotations will increase greatly complexity of each content instance and prevents a sophisticated semantic handling.

Modelling the semantics behind the data needs as much attention as modelling the structural properties of content. For that reason we propose a content data model which integrates structure, intension, usage, and origin on the same level. We start with the definition of content instances in this model.

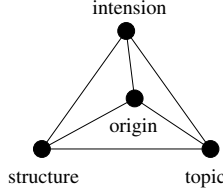


Figure 2. Dimensions of Content Chunks.

Definition 1. A content chunk is a tuple $\mathcal{C} = (D, I, T, O)$ where D is the structural representation of a content instance, I an intensional description, T a topic description expressing the usage of content, and O the specification of the context where the content instance is used. The state of a content system is a finite set of content chunks.

Figure 2 visualizes the four dimensions of content chunks. Each content chunk expresses the association identifying ‘what (structure) is used by whom (origin, context) in which way (topic) under which assumptions and thoughts (intension)’. In the following paragraphs we will refine these notions.

The Structure Dimension

The structural part of a content chunk reflects the classical notion of a content instance. Depending on the nature of the content data may be represented using an instance of a database schema formulated in ERM or UML, a semistructured resource such as a XML document, or a subsymbolic resource such as a picture.

Definition 2. Let \mathcal{L} be a set of (supported) modelling languages, \mathcal{S}_L the set of all schemata expressible with a certain modelling language $L \in \mathcal{L}$ and Σ_S the set of all possible states of a certain schema S . The structural component D of a content chunk \mathcal{C} is a triple (L, S, I) denoting a modelling language $L \in \mathcal{L}$, a schema $S \in \mathcal{S}_L$, and an instance $I \in \Sigma_S$.

In our example, $(\text{'HERM'}, s, i)$ is the structural part of a content chunk if s denotes the schema in Fig. 1 and i an instance which associates e.g., the entity type *List* with the entity set $\{\{No : 1\}\}$, the entity type *Product* with the set $\{\{No : 134, Descr : Book, Price : 16.99\}, \{No : 521, Descr : CD, Price : 9.95\}\}$, and the relationship type *consistsOf* with the relationship set $\{\{List.No : 1, Product.No : 134, pos : 1\}, \{List.No : 1, Product.No : 521, pos : 2\}\}$.

The structure dimension of content chunks is based on the theory of media types [20]. Media types [17] combine views and their functions into one type. Media types may be linked to other media types. For instance, we may distinguish input data for the workflow, retrieval data for the workflow, output data of the workflow, display data suites for each stage of the workflows, and escorting data supporting the understanding of each stage of the workflow.

Media objects may be structured, semi-structured, or unstructured by the media types. They are data that are generated from underlying databases, ordered, hierarchically representable, tailorable to various needs and enhanced by functionality for its usage. Since users have very different needs in data depending on their work history, their

portfolio, their profile and their environment media types are packed into containers. Containers provide the full functionality necessary for the application and use a special delivery and extraction facility. The media type suite is managed by a system consisting of three components:

Media object extraction system: Media objects are extracted and purged from database, information or knowledge base systems and summarized and compiled into media objects. Media objects have a structuring and a functionality which allows to use these in a variety of ways depending on the current task.

Media object storage and retrieval system: Media objects can be generated on the fly whenever we need the content or can be stored in the storage and retrieval subsystem. Since their generation is usually complex and a variety of versions must be kept, we store these media objects in the subsystem.

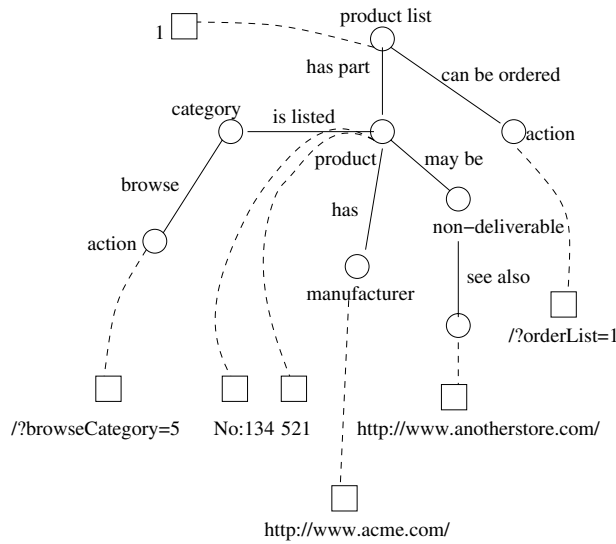
Media object delivery system: Media objects are used in a large variety of tasks, by a large variety of users in various social and organizational contexts and further in various environments. We use a media object delivery system for delivering data to the user in form the user has requested. Containers contain and manage the set of media object that are delivered to one user. The user receives the user-adapted container and may use this container as the desktop database.

This understanding closely follows the data warehouse paradigm. It is also based on the classical model-view-control paradigm. We generalize this paradigm to media objects, which may be viewed in a large variety of ways and which can be generated and controlled by generators.

The Topic Dimension

The topic part of a content chunk is the conceptual counterpart to the presentation facilities of content management systems. Available systems offer template mechanisms (e.g., based on XSLT or scripting languages such as PHP or JSP) which transform a content instance to a physical representation ready for delivery through an output channel, e.g., HTML Web pages, e-mails, or PDF documents. Instead of coding presentation on the level of rendering templates a more abstract approach should be used. Topic maps ([12,28]) provide the general data structure for a user-dependent view on content on the conceptual level. Expressing content via topic maps fulfills the following tasks during content delivery:

- The data structure is transformed to local vocabulary, e.g., according to a corporate identity or internationalization. In our example attribute names may be changed to language dependent labels. The prices of our products may be converted to local currencies or may be recalculated according to different tax regulations.
- The content is embedded into the usage context. The onion approach for a step-wise generation of delivered content ([25,26]) defines different kinds of embeddings depending on the profile of a user (characterization of the properties of the user such as language, skill level, or preferences) and portfolio (tasks which have to be, should be, and can be fulfilled by the user, see [7]). This information is obtained from the specifications of workflows and storyboards for interaction and



added to the topic map as supplementary content. There are different kinds of supplementary content:

- *static content*, e.g., the logo of the company or statically linked elements such as advertisement banners,
- *decorative content* which is dependent on the current usage context but has no meaning to the application such as contextual help or integrated services such as a contextual weather forecast,
- *additionally delivered content* such as information about manufactures or links to related products in our Web shop example, and
- *navigational events* such as navigational links allow the user to interact with the system.
- Multiple topic maps may be merged for multi-modal applications.

Supplementary content is incorporated in the topic map by parameterized queries on the set of content chunks and user specifications which characterize the occurrences of topics defined in the topic map. These queries are evaluated during content delivery and produce the topic map which can finally be rendered.

The topic part of a content chunk in our example may be the topic map depicted in Fig. 3. This topic map reflects our product list in the context which supplies additional information on these products. This topic map can be transformed to a physical representation (e.g., a HTML page) using the usual techniques mentioned above.

Topic parts of a content chunk thus serve for several purposes: to represent ideas or understandings; to represent a complex structure of content chunks and their related chunks; to communicate complexes of content chunks; to aid understanding by explicitly integrating new and old content chunks; and to assess understanding or diagnose misunderstanding.

The Origin Dimension

To express content syndication information about the origin of content has to be stored. The provenance of data was already studied on the instance level ([3,34,1]) especially for scientific data sets. We can adapt these results for our purposes. We choose a finite set \mathcal{C} from a universe $\mathcal{U}_{\mathcal{C}}$ of contexts. Each context in \mathcal{C} represents a point of view on the application area under consideration. These points of view may be different points of view of the same user or may belong to different users. Because all these contexts are views on the same universe of discourse they are related: data, intensions, and topics may be exchanged between contexts. Actions in one context may affect other contexts.

Definition 3. *Let $\mathcal{U}_{\mathcal{C}}$ be a universe of contexts and let $\mathcal{C} \subset \mathcal{U}_{\mathcal{C}}$ be a finite set of contexts. Further, let $\mathcal{A} = \{\mathfrak{A}_1, \dots, \mathfrak{A}_n\}$ be a set of content chunks. The origin part of a content chunk \mathfrak{C} is a tuple (c, \mathcal{A}) with a context $c \in \mathcal{C}$ where the content chunk \mathfrak{C} resides and a set \mathcal{A} of content chunks which are considered to be the ancestors of this chunk. The graph implied by this ancestor relationship between content chunks has to be acyclic.*

Connections between content chunks enable the exchange and transformation of data, intensions, and topics between different contexts. In our example we may define a content chunk representing our product list together with a topic map for rendering a shopping cart. By adapting the topic map as well as the intension we may construct a content chunk which renders an order confirmation.

The origin dimension also reflects the purpose of the content chunk. Content chunks have a function such as to give an instruction, to control behaviour, to transmit ideas in a memorable form, to enhance social cohesion, or to give users a better understanding. For example, a content chunk that reflect a piece of knowledge may start with a mystery that leads to a conflict situation, may continue with an explanation of the solution or of the discovery as the turning point and may conclude with a resolution of the conflict situation. Context [14] injection must thus be an integral element for content chunk processing.

Our origin model for content chunks extends the usage thesis of [33] that mainly reflect the communication act between a sender and receiver with their intentions, backgrounds, cultures and relationships. Usage context should also consider excluded receivers, value of content chunks to receivers, groups or societies. Content chunks are thus generically [2] enhanced by context, refined by intended specifics and instantiated by their specific usage.

The Intension Dimension

The intention dimension is based on concepts. They are the building blocks in human thinking and reasoning, and as such highly flexible. They can be general or specific, concrete or abstract, natural or technological, artistic or scientific. They can apply to things that are real or imaginary. They provide a help for distinguishing between things we observe in the world, or ideas such as truth and falsity, appearance and reality and continuity and discontinuity. Abstract concepts are useful for characterisation of observations, thoughts and expressions. Typical abstract concepts are truth and falsity, sameness and difference, wholes and parts, subjectivity and objectivity, appearance and reality, conti-

nuity and discontinuity, sense and reference, meaningful and meaningless and problem and solution. They govern different kinds of human thinking at a fundamental level.

Concepts are vital to the efficient functioning of semantic Wikis. They are organised bundles of stored knowledge which represent an articulation of events, entities, situations, and so on experience. Concepts are necessary for an understanding, for the organisation, for sharing Wikis and for communication. We may assume a simple association between the components of Wikis and concept. The associations may form a complex multi-dimensional network. They may be of specific types such as *kind-of*, *is-part-of*, *is-used-for* and of variable strength. Associations typically correspond to concepts of a more schematic kind than the concepts which they serve to connect.

The classical approach to concepts is based on description of necessary and sufficient criteria for content-concept association. We notice however that most concepts characterising content chunks cannot be captured by means of a set of necessary and sufficient features. Many natural concepts are fuzzy and contextually flexible. Therefore we need to extend the approaches typically assumed for formal semantics to natural semantics. Additionally, the association of content to concepts must not be strict. Some content may be a better example to a concept than other content.

The prototype approach for concept-content association is also limited. Ratings or selections of prototypes are strongly context dependent, e.g., culture dependent and actor dependent. Prototypes are given with certain preference, frequency, sample extraction, learning background, level of verifiability, and under time pressure. The degree of association may vary over time, may be dependent on the concrete usage, and bound by the representation language chosen. Prototype content may also be more or less specific or general for concepts.

Concepts are typically expressed through propositions. The meaning has typically two parts: an element of assertion and something that is asserted. What is asserted is called proposition. The simplest type of proposition consists of an argument and a predicate. Semantical units or propositions are interrelated by entailment. Entailment is different from material implication and relates propositions by forward propagation of truth and backward propagation of falsity. Propositions can be contraries, contradictories, or independent. They may belong to a category or genre of expression, are given in a certain style or manner, are often based on stereotypical norms of expression, depend on ideas and values that are employed to justify, support or guide the expression, reflect aspects of culture or social order, are shaped according to the community that uses them, and are configured by theories or paradigms.

We also may distinguish between the existential approach to meaning based on a correlation of expressions in a language with aspects in the world. The intentional approach associates some kind of representation with concepts as the main constituents of the sense and depends on the cultural context. Whenever content is difficult to interpret then we need to consider concepts, deep structures, unconscious foundations, hidden symbols, annotations or underlying pattern supporting it. If content seems to transparent then we do not need to look for these things. It is often surprising how much background information is necessary for understanding content even such content that appear on the surface to be wholly transparent. There are various connotations and denotations that content may have. We have to consider the arrangements and laws for constructing content phenomena (langue) as well as the various instances that are constructed by constructors and laws (parole). Content can be coded in various ways, e.g. based on different

representation such as text or multimedia elements. Content can be differently categorized and organised. We may use conventions that draw on common forms of knowledge. Furthermore, we need to devise different ways for understanding and for association of concepts to content.

The intension of a content chunk expresses the purpose of the content as well as meanings and thoughts about the content chunk. Thoughts about some object may be expressed using a general description frame. A sophisticated and generic frame was given by Zachman in the context of specifications in software engineering ([35,23]): each thought is expressed by formulating the facets *who*, *what*, *when*, *where*, *how*, and *why*. Each facet is specified by a concept. A concept itself is a small logical theory. We base our notion of concepts on intensional logics, especially on a restricted version of *Transparent Intensional Logic* (TIL) introduced by Tichý ([6]). TIL introduces the notions of intensional functions which map modalities (time, place, object identity, possibility, etc.) to values and intensional constructions building the intension of more complex expressions out of its components.

In our example we may introduce the concepts of *customers*, *products*, *shopping carts*, *wish lists*, *product orders*, etc. The concept *shopping cart* implies an intension of what a shopping cart is: it is a list of products selected from the offers in our Web shop. These products may be purchased in the future.

TIL analyses the intension of a concept down to the objectual base (calculating valuations of the intension behind the sentence ‘Products are associated with a description and a price’ considers all possible valuations of *product*, *description*, *price*, *associated with* and even *and* in ‘one shot’.) This is not the natural way of thinking. We modify the TIL approach in the following way:

- We introduce different types of individuals in the objectual base. TIL defines a single class ι of individuals. Introducing multiple (disjunct) classes ι_1, \dots, ι_n together with operations and predicates (such as ordering relations) corresponds to the definition of data types for attributes in classical database modelling. As defined in TIL there is at least one class o of truth values (*true*, *false*) with the usual operations and predicates. The intension behind these operations and predicates is no longer transparent in terms of TIL.
- We support different types of modalities. TIL is specialized on modalities *object identity* and *time* and defines each intensional function on these modalities. Because there are other modalities (especially the *possibility* of a fact) and some intensions may be expressed in a more compact way if e.g., the time modality is omitted we will define intensional functions over arbitrary modalities from a given universe Ω of modalities.
- The objectual base consists of all first order types defined in TIL:
 - ι_i , o , and ω_i are first order types,
 - each partial function $\alpha_1 \times \alpha_k \rightarrow \beta$ with first order types α_i and β is a first order type,
 - nothing else is a first order type.

Definition 4. An intensional function is a function $f : \omega_{i_1} \times \omega_{i_k} \rightarrow \alpha$ mapping possible worlds $(w_{i_1}, \dots, w_{i_k})$ to instances of a first order type α . An intensional func-

tion is called non-trivial if there are two possible worlds (w_1, \dots, w_k) , (v_1, \dots, v_k) with $f(w_1, \dots, w_k) \neq f(v_1, \dots, v_k)$.

All first order types which are no intensional functions are called extensions.

Intensional functions can be used to express the usual type constructors: classes can be represented by their characteristic function, attributes by functions mapping to individuals, associations between objects by functions mapping to object identities.

In contrast to TIL we consider different kinds of intensional functions. The first kind is defined in a non-transparent way. Typical examples are extensional objects such as operations and predicates on the objectual base. Other non-transparent functions may be obtained from external data sources. For example, the concept of a *customer* may be represented as a characteristic function over modalities ω (object identity) and τ (time): $isCustomer : \omega \times \tau \rightarrow o$. The valuation of this function may be determined by coupling the function with the customer database of our Web shop: $isCustomer(w, t) = true$ if and only if the object with identifier w is registered as a customer in our customer database at time t . Non-transparent intensional functions may be evaluated but do not reveal the internal structure of the valuation or their relationship to other intensional functions.

The second kind of intensional function is built in a transparent way: an intensional construction is used to relate valuations of the function with valuations of other first order types. Tichý introduced four types of constructions: variables of type α , trivialization (using objects in constructions), composition (application of values to a function) and closure (creation of functions).

Definition 5. We consider a single context $c \in \mathcal{C}$. We organize intensional functions on strata in the following way:

- Operations and predicates on the objectual base (such as boolean connectives) as well as all non-transparent intensional functions and all intensional functions imported from contexts other than c are considered to be functions on stratum 0.
- Let k be an intensional construction with free variables x_i and a total mapping $p : \mathcal{X} \rightarrow \mathcal{F}$ from variables $\mathcal{X} = \{x_1, \dots, x_n\}$ to intensional functions $\mathcal{F} = \{f_1, \dots, f_m\}$ where the stratum of f_j is at most $s - 1$. The intensional function constructed by k is considered to be a function on stratum s .

The layering of intensional functions implies the independence of intensional functions on lower strata from intensional functions on higher strata and especially from their usage in constructions. This enables the determination of valuations of intensional functions on higher strata by first fixing the valuations of intensional functions on lower strata. This restricts expressiveness with respect to TIL. The strict monoton layering may be relaxed to constructions out of functions from the same stratum. Functions can be lifted to higher strata by using identity constructions, so we will allow the direct assignment of functions to strata higher than given by the definition.

Intensional constructions represent the terminological knowledge in traditional ontologies. Constructors such as ‘is-a’, ‘part-of’, or ‘union-of’ represent a fixed, preselected, and not configurable set of intensional constructions.

Building intensions by intensional constructions does not associate valuations of this intensional function with concrete objects. Beside intensional (T-box) reasoning based on constructions, properties of valuations of intensional functions have to be revealed.

Definition 6. Let c be a context, $\mathcal{F} = \{f_1, \dots, f_n\}$ a set of intensional functions, L a logical language and \mathcal{T} a theory with sentences from L formulating the knowledge about the valuations of \mathcal{F} with respect to the layering of intensional functions in c . The tuple $\mathfrak{B} = (\mathcal{F}, \mathcal{T})$ is called a concept in context c .

In our Web shop example we might consider intensional functions $isCustomer : \omega \times \tau \rightarrow o$, defined in a non-transparent way as mentioned above. Assuming that a customer will remain to be a customer for all the time we can express this in our small theory about customers:

$$isCustomer(w, t) \implies (\forall t' > t)(isCustomer(w, t'))$$

In another example shopping carts ($isShoppingCart : \omega \times \tau \rightarrow o$) might become an order list ($isOrderList : \omega \times \tau \times \eta \rightarrow o$ for possibilities η):

$$isShoppingCart(w, t) \implies (\exists n' \in \eta, t' \in \tau)(isOrderList(w, t', n'))$$

With the definition of concepts we can finally construct content intensions:

Definition 7. Let \mathfrak{S} be a set of facets (e.g., according to the Zachman framework). A content intension is a set of functions $i : \mathfrak{S} \rightarrow \mathcal{B}$ mapping facets from \mathfrak{S} to concepts from $\mathcal{B} = \{\mathfrak{B}_1, \dots, \mathfrak{B}_n\}$ in the current context.

3. Query Facilities for Content Chunks

The definition of content chunks as combinations of data, intension, topic, and origin enables several kinds of query facilities in content management systems. In the rest of the paper we use \mathcal{D} for the set of all structure definitions in the state of the CMS, \mathcal{I} for the set of all defined intensions, \mathcal{T} the set of all defined topic maps, and \mathcal{C} for the set of all contexts. Typical examples for query functions are:

- Structural queries remain unchanged. Depending on the modelling language(s) the usual query facilities are present, e.g., return all products from a certain product list.
- The function $explain : \mathcal{D} \rightarrow 2^{\mathcal{I}}$ returns all intensions associated with a certain data instance.
- $sample : \mathcal{I} \rightarrow 2^{\mathcal{D}}$ returns all data instances associated with a certain intension.
- $express : \mathcal{D} \times \mathcal{I} \times \mathcal{C} \rightarrow 2^{\mathcal{T}}$ returns all topic maps associated with the given data object under the given intension in the given context.
- $understand : \mathcal{T} \times \mathcal{C} \rightarrow 2^{\mathcal{I} \times \mathcal{D}}$ returns data instances together with an intension for the given topic map and context.
- $find : \mathcal{C} \rightarrow 2^{\mathcal{C}}$ returns the contexts which incorporated content from this context.
- T-box reasoning in a generalized fashion is available by evaluating the intensional constructions. There is additional reasoning support, as depicted in Fig. 4. Properties of a concept relevant within a context are expressed in small local theories. We do not assume that this theory is a complete description of the concept but reveals relevant aspects. Concepts may be imported by other con-

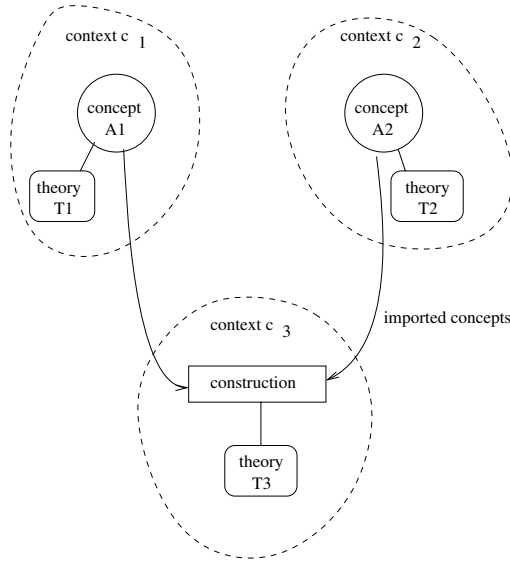


Figure 4. Imported Concepts, Constructions, and Local Theories.

texts while possibly different properties may become important. This is expressed by associating a different theory to the corresponding intensional function. For example, in a certain context it may be important to have a conception about the time when a person became a customer. An additional intensional function $customerRegistrationDate : \omega \rightarrow \tau$ may be introduced on a stratum lower than $isCustomer$ while the local theory of the concept *customer* is enhanced by the constraint

$$(\forall w \in \omega)(\forall t < customerRegistrationDate(w)) \\ (isCustomer(w, t) = false)$$

Evaluation of properties follows this construction strategy:

- First, the theory locally defined within the current context is used to prove the desired property.
- If the local proof was not successful, the intensional construction is investigated and reasoning is delegated to original contexts where the concept was imported from.

It is obvious that reasoning in this fashion does not ensure decidability but enables the delivery of precalculated relevant aspects which may not be accessible by pure intensional reasoning.

4. Content Schemata

In Section 2 we defined the building blocks of content as arbitrary tuples (D, I, T, O) . Considering typical application scenarios of content management systems arbitrary associations can be restricted to support additional content management functionality:

- There are relationships between intensions and structural properties. Reasoning about intensional properties is reflected by certain values of the associated data instances. For example, reasoning about prices should be reflected by appropriate attributes in the structural definition. Non-transparently defined intensional functions must be directly computed from data.
- Information expressed in a topic map should be related to the underlying data and vice versa.
- Information can only be expressed or understood if there is an appropriate intension. On the other side, every intension should be expressible.
- Content which is imported from different contexts may not be completely revised but transformed.
- Not every intensional construction should be allowed. To restrict complexity a configurable set of construction templates has to be defined which incorporates the conceptual theories from the sources to build theories in the target context.

Restrictions may be expressed by constraint relations between the four dimensions of content chunks. To support content management functionality a mapping approach is better. There are three general tasks which have to be fulfilled during content management: content is created, selected content is delivered to the user, and content is exchanged between different contexts. Content creation in terms of our data model is the mapping of a topic map within a context to combinations of an intension and a data instance. Content delivery is the mapping between a data instance and an intension within a context to a topic map. Content translation maps content chunks from one context to another.

Definition 8. A content schema is a tuple $(generate, deliver, exchange)$ with a function $generate : \mathcal{T} \times \mathcal{C} \rightarrow 2^{\mathcal{D} \times \mathcal{I}}$, a function $deliver : \mathcal{I} \times \mathcal{D} \times \mathcal{C} \rightarrow \mathcal{T}$, and a function $exchange : \mathcal{D} \times \mathcal{I} \times \mathcal{T} \times \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{D} \times \mathcal{I} \times \mathcal{T} \times \mathcal{C}$.

These functions are defined for each context separately. First, a set of base intensions is defined. These base intensions rely on concepts (such as *customer* or *shopping cart*) which may be defined transparently or non-transparently. These base intensions are associated with a data schema (L, S) (where L is a modelling language and S is a schema expressed in this language), a topic map template incorporating the data by associated data queries and a data template defining the data instance by queries on the topic map.

Definition 9. Let $\{k_1, \dots, k_n\}$ be a set of intensional constructions. An intensional construction template is a tuple $(\{k_1, \dots, k_n\}, p, m)$ with intensional constructions k_i for each facet in the intension specification frame, a parameter assignment specification $p : \mathcal{X} \rightarrow \mathcal{B}$ mapping variables from $\mathcal{B} = \{x_1, \dots, x_k\}$ to concepts from $\mathcal{B} = \{\mathfrak{B}_1, \dots, \mathfrak{B}_l\}$ restricting valuations for variable substitutions in $\{k_i\}$ to the given concepts, and a merging function m which creates

- the logical theory T out of the theories associated to \mathcal{X} ,
- the data schema out of the data schemata of \mathcal{X} ,
- the topic map template out of the topic map templates of \mathcal{X} , and
- the data template out of the data templates of \mathcal{X} .

The definition of the data schema, the topic map template, and the data template implies the content generation and content delivery functions. The creation of the logical theory out of other concepts is given by a compatibility relation between models of these theories as defined by the Local Model Semantics framework ([8]).

5. Semantic Wikis: Enabling Collaborative Content Annotation and Foundation

Communities form an interacting group of various actors in a common location, common intension, and common time. They are based on shared experience, interest, or conviction, and voluntary interaction among members with the intension of members welfare and collective welfare. They can have more or less structure and more or less committed members.

A wiki is a collaborative web site set up to allow user editing and adding of content by any user who has access to it. Wikis have changed access and habits of internet users. The right and wrong usage of wikipedia is already widely studied in literature, e.g. the journal *First Monday* provides a detailed study of Web 2.0 in issue 13, March 2008 and of wiki misuse in more than a three-score papers in various issues. The main functionality provided for wikis is

- management of content chunks with their structure, intention, origin and topic,
- annotation management for users depending on their role, rights and obligations,
- explanation, exploration and knowledge elicitation and gathering support for readers of content chunks,
- presentation of information in a variety of ways and for a variety of environments,
- user management depending on the roles, functions and rights a user may have on the content,
- security and safety management for integrity of content and information, and
- history and evolution management that allows to show the development of the wiki and to restore previous versions.

We concentrate the remaining part of the paper to the first three main functionalities of wiki systems. These three functionalities are backed by our approach to *semantic wikis*. Presentation may also include generic adaptation to the user environment and features for marking content [15]. Wiki systems should integrate features that have been developed for customer management. Wikis are generally designed with a functionality that makes it easy to correct mistakes. Since this functionality is a target for attacks on content and on the system, wiki systems are extended by security and quality management features. Thus, they provide a means to verify the validity of recent additions, changes, corrections, replacements etc. to the content. History and development information can be maintained through docketts [27] and the diff feature that highlights changes between two revisions. Wiki systems are special web information systems. They support information seeking life cases [5,21], are based on storyboards for creation and consumption of information [17] and require a sophisticated user profile and portfolio model [19].

Wiki systems share and encourage several features with generalized content management systems, which are used by enterprises and communities-of-practice. They are maintained, developed and enriched by communities of leisure, interest or practice. Community members are allowed to instantly change the content (usually Web pages.) There

are some minimal requirements to the content chunk for wikis. The name or annotation of a content chunk is typically embedded in the hyperlink and interlinked with other chunks. Content chunks can be partially created or edited at anytime by anyone (with certain limitations for protected articles). They are editable through the web browser. Their evolution history is accessible through a history/versioning view, which also supports version differencing (“diff”), retrieving prior versions and summary of most recent additions/modifications. Additionally, easy revert of changes is possible. We can extend this conception of Wikis and look forward on how functionality of Wikis may evolve by incorporating topically annotated and intensionally founded content.

Semantic wikis¹ enhance content that is displayed in the web with fully considered and perfected concepts or verified knowledge and with user annotation or topics. It thus formalises the notion of wikis enhanced by ontologies [13], clarifies the knowledge basis and provides a basis for using data from the web in a form that corresponds to the user demands, their information portfolio and their personal profile.

Using Communities for Content Generation

Content creation and content annotation are resource-intensive processes. Introducing a user- and usage-centric approach to content handling as presented in this paper, these processes can be distributed through a social network, adapting the notions of the Web 2.0 initiative. One of the most severe requirement to wiki evolution is trustworthiness of the wiki. Everybody who uses wiki systems such as wikipedia observes that the competence level, the education profile, the work portfolio, the biases, the intensions (e.g., trolling) and the psychographical level of wiki creators has a huge impact on the quality of a wiki. Wikis that can be created by almost everybody are typically of lower quality than those that can only be created by experts in the area. As an example we accessed the entry ‘Dresden’ in wikipedia.org. In the average, each second sentence was not completely correct.

Wiki systems are considered to be special content management systems which allow the user to instantly change the content (usually Web pages.) We can extend this notion to semantically enriched content:

- Content may be loaded into the system. This reflects the usual process of editing pages in a Wiki. The load process results in stored data instances in the CMS which can be extracted via search templates or associated with metadata in the usual sense (e.g., editorial parameters).
- Data instances may be associated with intensional descriptions such as copyrights and access rights.
- The user may annotate the content after searching for it, e.g., making recommendations on products in our Web shop example. A recommendation can be expressed by an additional intension on the content chunk expressing that the current user interprets the data as a product recommendation. The local theory expresses the fact, that this user has bought these products or might buy these products in the future.

¹Our concept of semantic wikis should not be mistaken as a concept of Web 3.0. Web 3.0 or semantic web aims in annotation of content on the basis of an ontology which is commonly accepted by a community. Proposals such as the Semantic MediaWiki add a fixed annotation to concepts similar to tagging in websites such as delicious.us.

- Another user may explore the notion of a ‘recommendation’ from the context of the first user if he sees the same data instance and looks for associated intensions. Afterwards, this user may use this concept to annotate other data instances.
- Users may refine the local theory of a concept to incorporate knowledge which was hidden so far.
- Users may associate new topic maps to content to create different (e.g., localized) versions.

Modelling Wiki Functionality Based on Content Chunks

Beside supporting content generation and annotation by social networking, semantically and user-specifically enriched content chunks are the base for modelling collaboration within a network of users. Collaboration is seen ([18,32]) as a process of interactive knowledge exchange by several people working together towards a common goal. Collaboration can be characterized ([25]) by three facets: *communication*, *cooperation*, and *coordination*. The communication facet defines the exchange protocols of content between users. The cooperation facet relies on the workflow of the collaboration by specifying *who* (actor) has to deliver *which* results to *whom*. The coordination facet defines the task management and synchronization between the collaborating partners to fulfill the cooperation goals.

Collaboration in social networks is usually defined in an implicit and decentralized way, so classical workflow management systems with fixed process definitions cannot be applied. The content data model defined in this paper can be used to annotate content with the specified collaboration frame to express

- the history of content and content changes,
- the purposes of content changes and content usage,
- future tasks on content chunks and therefore,
- access rights on content chunks.

In the context of collaboration the specification of users becomes important. Conceptually, users are handled by a set of concepts \mathcal{A} called *actors* (such as administrator, moderator, registered user, unexperienced user, guest user, etc.) Actors define roles of users and therefore imply a grouping on the set of users. According to our definition of concepts each actor is associated with a small logical theory expressing the properties which are common to all users in the user group of the actor.

Communication between users takes place by topics. Topic map fragments have to be defined in the content schema to express tasks in the collaboration frame characterized above. Figure 5 shows an example for a topic map concerning a product within our Web shop. A typical task which can be done by users is to write a comment. The topic map for expressing the product incorporates only comments fulfilling the intension of a *proofread comment*. To write a comment the topic map is merged with a topic map requesting comments from users. Occurrences of these topics are linked with dialog specifications that offer access to the CMS services to fulfill the desired task.

These topic map fragments which express task specifications are associated with an intension according to our collaboration frame (who wrote a comment on which product.) in the content schema. Coordination is done by expressing obligations (e.g., adoptions of [29]) on the content chunk in the local theory of the intension, e.g., a moderator

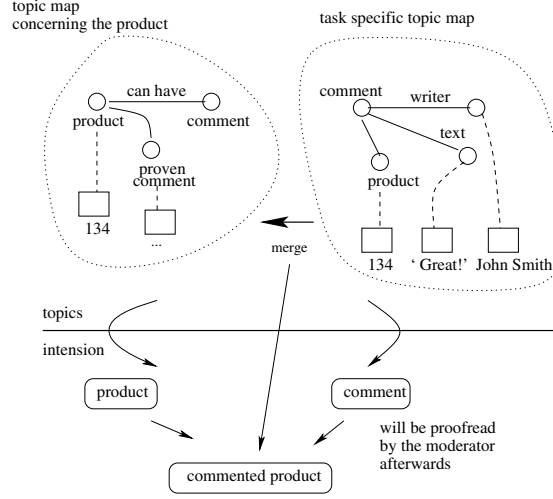


Figure 5. Topic Map with Task Specification.

has to proofread the comment after the comment was written and before the comment is published. For that reason there is a topic map defining the proofreading task for moderators which can be merged with the topic map associated with the intension of *commented products*. This merging process creates the intension of a *proofread comment*, characterized by the fact in the local theory that at a time point t the proofreading task took place:

$$\begin{aligned} isProofReadComment(w) &:= \\ (\exists w', t < now) &(moderator(w, t) \wedge proofread(w, w')) \end{aligned}$$

From Wikis to Semantic Wikis: Extended Functionality

The new query functionality on content with topic annotations and intensional foundations enables additional high-level functionality for content management systems in social environments to support more sophisticated content management services:

- Sophisticated and automatically generated graphical user interfaces such as WYSIWYG editors rely on a user-centric topic-based content annotation to provide information in the right fashion at the right time.
- Community services such as contributions to communities, joining communities, meetings, publications, or community organization as well as their integration can be intensionally modelled.
- Online collaboration support active discussion and interaction among participants as well as content exchange.
- Content changes within a collaborating environment may be conflicting. Expressing the purposes of changes may help to solve these conflicts.
- Task annotations support modelling of interaction scenarios and coordination facilities such as schedules to enable project management functions.
- Secretarial functions such as filtering or ordering can be intensionally expressed and enforced by appropriate topic map definitions.

- Blackboard facilities support tracing of tasks, members, schedules, and documents. Report functions may be incorporated for non-members of the community.
- Ad hoc (and implicit) communities are supported. Members can conduct their own communities, interests, tasks, and portfolios by defining private workspaces.
- Asynchronous as well as synchronous collaboration is supported depending on the handling of annotations and intensions.

6. Conclusions

In this paper we are introducing a data model for content management systems which handles content as associations between the data itself, the intension behind the data, the usage of data and the origin of data. Content annotation is treated according to its purpose: terminology which is common sense in a community is shared in ontologies. Concepts which are only relevant to certain users or in certain situations are defined locally and removed from the global ontologies to make reasoning about global terminology easier. Local concepts may be exchanged and adapted between different usage contexts. For that reason concepts are seen not only as notions from an ontology but as small logical theories. Additionally, intensional annotation is separated from usage annotation to allow different expressions of the same data under the same intension.

Because of the reduced size of annotation ontologies, the local definition of concepts and the suggested evaluation strategies according to the origin definitions of the content, the separation of concerns within the data model allows a better automated reasoning support than simple (S, P, O) annotation frameworks although decidability as well as soundness and completeness of the reasoning process cannot be guaranteed. The user-centric approach together with the facility of explicitly incorporating and exchanging hidden knowledge into local theories behind a concept ensure the usability within known application scenarios when automated reasoning fails. Adding local and user-specific semantics to content chunks is a prerequisite for distributing content over social networks and therefore extends current Web 2.0 technologies in a natural way. While today Wikis support open communities mainly interested in free-form changes of content, Semantic Wikis may also support transaction oriented communities with the need of at least partially controlled collaboration.

References

- [1] O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. ULDBs: Databases with Uncertainty and Lineage. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 953–964, September 2006.
- [2] A. Bienemann, K.-D. Schewe, and B. Thalheim. Towards a theory of genericity based on government and binding. In *Proc. ER'06, LNCS 4215*, pages 311–324. Springer, 2006.
- [3] P. Bunemann, S. Khanna, and W.-C. Tan. Data Provenance: Some Basic Issues. *Foundations of Software Technology and Theoretical Computer Science*, 2000.
- [4] Dublin Core Metadata Initiative. Dublin Core. <http://dublincore.org/>, June 2007.
- [5] A. Düsterhöft and B. Thalheim. Linguistic based search facilities in snowflake-like database schemes. *Data and Knowledge Engineering*, 48:177–198, 2004.
- [6] M. Duží and P. Materna. Constructions. http://til.phil.muni.cz/text/constructions_duzi_materna.pdf, 2000.
- [7] G. Fiedler, A. Czerniak, D. Fleischer, H. Rumohr, M. Spindler, and B. Thalheim. Content Warehouses. Preprint 0605, Department of Computer Science, Kiel University, March 2006.

- [8] C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. <http://citeseer.ist.psu.edu/481285.html>, April 2000.
- [9] V. Haarslev and R. Möller. Racer: An OWL Reasoning Agent for the Semantic Web. In *Proceedings of the International Workshop on Applications, Products and Services of Web-based Support Systems, in conjunction with the 2003 IEEE/WIC International Conference on Web Intelligence, Halifax, Canada, October 13*, pages 91–95, 2003.
- [10] V. Haarslev, R. Möller, and M. Wessel. Description Logic Inference Technology: Lessons Learned in the Trenches. In I. Horrocks, U. Sattler, and F. Wolter, editors, *Proc. International Workshop on Description Logics*, 2005.
- [11] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.
- [12] ISO/IEC JTC1/SC34. Topic Maps – Data Model. <http://www.isotopicmaps.org/sam/sam-model/>, June 2006.
- [13] C. Kaliszyk, P. Corbineau, F. Wiedijk, J. McKinna, and H. Geuvers. A real semantic web for mathematics deserves a real semantics. In *Proceedings of the 3rd Semantic Wiki Workshop (SemWiki 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, volume 360 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [14] R. Kaschek, K.-D. Schewe, B. Thalheim, and L. Zhang. Integrating context in conceptual modelling for web information systems, web services, e-business, and the semantic web. In *WES 2003, LNCS 3095*, pages 77–88. Springer, 2003.
- [15] T. Moritz, K.-D. Schewe, and B. Thalheim. Strategic modeling of web information systems and its impact on visual design patterns. In F. Frasincar, G.-J. Houben, and R. Vdovjak, editors, *WISM'05*, pages 5–13, Sydney, 2005.
- [16] L. Reeve and H. Han. Survey of semantic annotation platforms. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1634–1638, New York, NY, USA, 2005. ACM Press.
- [17] K.-D. Schewe and B. Thalheim. Conceptual modelling of web information systems. *Data and Knowledge Engineering*, 54:147–188, 2005.
- [18] K.-D. Schewe and B. Thalheim. Development of collaboration frameworks for web information systems. In *IJCAI'07 (20th Int. Joint Conf on Artificial Intelligence, Section EMC'07 (Evolutionary models of collaboration)*, pages 27–32, Hyderabad, 2007.
- [19] K.-D. Schewe and B. Thalheim. Pragmatics of storyboarding for web information systems: Usage analysis. *Int. Journal Web and Grid Services*, 3(2):128–169, 2007.
- [20] K.-D. Schewe and B. Thalheim. Facets of media types. In *Information Systems and e-Business Technologies, LNBIP 5*, pages 296–305, Berlin, 2008. Springer.
- [21] K.-D. Schewe and B. Thalheim. Life cases: A kernel element for web information systems engineering. In *WEBIST 2007, LNBIP 8*, pages 139–156, Berlin, 2008. Springer.
- [22] J. F. Sowa. *Knowledge Representation, Logical, Philosophical, and Computational Foundations*. Brooks/Cole, a division of Thomson Learning, Pacific Grove, California, 2000.
- [23] J. F. Sowa and J. A. Zachman. Extending and Formalizing the Framework for Information Systems Architecture. *IBM Systems Journal*, 31(3):590–616, 1992.
- [24] B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000. See also <http://www.is.informatik.uni-kiel.de/~thalheim/HERM.htm>.
- [25] B. Thalheim. Co-Design of Structuring, Functionality, Distribution, and Interactivity of Large Information Systems. Technical Report 15/03, Brandenburg University of Technology at Cottbus, 2003.
- [26] B. Thalheim. The conceptual framework to user-oriented content management. *Information Modelling and Knowledge Bases*, XVII:30–49, 2007.
- [27] B. Thalheim. Engineering database component ware. In *TEAA'06 post proceedings, LNCS 4473*, pages 1–15, Berlin, 2007. Springer.
- [28] TopicMaps.org. XML Topic Maps. <http://www.topicmaps.org/xtm/>, Sep 2006.
- [29] F. Voorbraak. The logic of actual obligation. An alternative approach to deontic logic. *Philosophical Studies*, 55, Issue 2:173–194, 1989.
- [30] W3C. Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/>, Feb 2004.
- [31] W3C RDF Core Working Group. Resource Description Framework (RDF). <http://www.w3.org/RDF/>, 2004.
- [32] Wikipedia. Collaboration. <http://en.wikipedia.org/wiki/Collaboration>, June 2007.
- [33] L. Wittgenstein. *Philosophical Investigations*. Basil Blackwell, Oxford, 1958.

- [34] A. Woodruff and M. Stonebraker. Supporting Fine-grained Data Lineage in a Database Visualization Environment. In *ICDE*, pages 91–102, 1997.
- [35] J. A. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 38(2/3):454–470, 1999.

Center Fragments for Upscaling and Verification in Database Semantics

Roland HAUSSER

Universität Erlangen-Nürnberg, Abteilung Computerlinguistik (CLUE)
rrh@linguistik.uni-erlangen.de

Abstract. The notion of a fragment was coined by Montague 1974 to illustrate the formal handling of certain puzzles, such as *de dicto/de re*, in a truth-conditional semantics for natural language. The idea of a fragment is methodological: given the vastness of a natural language, one begins with a system which is of limited data coverage, but formally explicit and functionally complete relative to a certain goal or standard.

Once a small fragment has been defined, there arises the task of *upscaling*, such as adding the treatment of another puzzle. Unfortunately, however, upscaling may turn out to be difficult or even impossible, depending on the assumptions of the initial fragment and the standard of functional completeness. For example, despite half a century of intensive research there is still no coherent formal account of a single natural language, verified computationally as nearly complete.

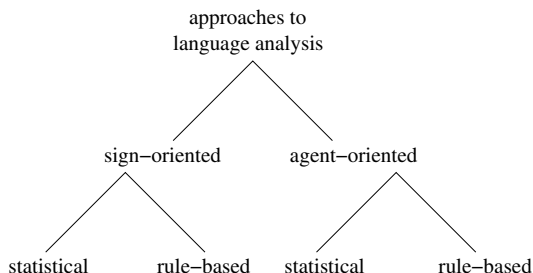
This paper proposes a new kind of fragment, called *center fragment*, designed to ensure the longterm success of upscaling. Its functional standard is the model of natural language communication of agent-oriented Database Semantics (DBS), based on the algorithm of time-linear LA-grammar and the data structure of prop-lets. Its language data are selected to represent the primary semantic relations of natural language, namely (a) functor-argument structure and (b) coordination, in their most basic form. The approach is illustrated with applications to four languages with distinctively different word orders, namely English, German, Korean, and Russian.

1. Sign-Oriented vs. Agent-Oriented Grammar

Of the many different ways to analyze natural language, the most basic distinction is between sign-oriented and agent-oriented approaches.¹ These may in turn each be divided into statistical and rule-based methods. For example, a sign-oriented approach based on statistics is corpus linguistics (e.g., Kučera and Francis 1967), a rule-based sign-oriented approach is the analysis of sentences within Nativist generative grammar (e.g., Chomsky 1965), an agent-oriented approach based on statistics is current work in robotics (e.g., Roy 2003), and a rule-based agent-oriented model is Database Semantics (e.g., NLC'06):

¹A related distinction is that between *language as product* and *language as action* of Clark 1996. However, while Clark counts Speech Act Theory (Austin 1962, Grice 1965, Searle 1969) among the language as action theories, it is nevertheless sign-oriented insofar as Speech Act Theory is based on enriching the analyzed language sign with performative clauses, such as *I declare*, *I promise*, etc. For a more detailed review of Ordinary Language Philosophy see FoCL'99, p. 84 f.

1.1. Basic Approaches to the Analysis of Language



While the statistical method is based on frequency, the rule-based method models functions or structures the importance of which may be in inverse proportion to their frequency.

The two rule-based approaches indicated in 1.1 are best distinguished in terms of their components. In a sign-oriented approach, the components are as follows:

1.2. Components of Grammar in a Sign-Oriented Approach

lexicon: list of analyzed words

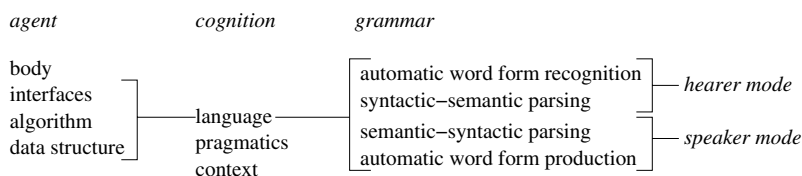
morphology: rule-based analysis of word forms

syntax: rule-based composition of word forms into sentences

semantics: rule-based interpretation of sentences

In an agent-oriented approach, the components of grammar are reorganized into the *speaker mode* and the *hearer mode*, and are embedded into a cognitive agent with language:

1.3. Components of a Talking Agent in Database Semantics



At the highest level of abstraction, the agent consists of a (i) body with (ii) external interfaces for recognition, e.g., vision and hearing, and action, e.g. locomotion and manipulation, an internal database with a suitable (iii) data structure, and an (iv) algorithm for reading content into and out of the database.

The content stored in the database is called the *context* and its processing is called the pragmatics. In agents with language, the context component is complemented by a language component,² designed as a grammar with the purpose of modeling natural language communication as a computer program in a talking robot. This functionality of the agent-oriented approach profoundly changes the ontological, empirical, and

²For a detailed description see NLC'06, Chapters 1–3.

methodological foundations of linguistic research as compared to the sign-oriented approaches.

First, there is a procedural notion of basic meaning: instead of the truth conditions of signs, meanings are defined in terms of the concepts used for realizing recognition and action. Second, the components of the grammar must be connected to the external interfaces of the agent, and functionally integrated into the speaker mode (recognition) and the hearer mode (action). Third, there must be a computationally well-defined interaction between the levels of language and context, such that reference is based solely on the procedures of cognition – without postulating any external relation between the sign and its referent(s) in the world.³

2. Coding Contents as Sets of Proplets

For representing content, Database Semantics combines the traditional notion of a proposition with a modern data structure designed for indexing, storage, and retrieval: content is coded as a set of non-recursive feature structures, called *proplets*.⁴ There are three basic kinds of proplets, called nouns, verbs, and adjectives, which are combined into basic propositions. In the following examples, each set of proplets is preceded by a paraphrase which characterizes the content with an equivalent expression of English:

2.1. Propositions with One-, Two-, and Three-Place Verbs

1. The girl dreams.

[sur:	[sur:
noun: girl	verb: dream
fnc: dream	arg: girl
prn: 1	prn: 1

2. The man sees the girl.

[sur:	[sur:	[sur:
noun: man	verb: see	noun: girl
fnc: see	arg: man girl	fnc: see
prn: 2	prn: 2	prn: 2

3. The man gives the girl a flower.

[sur:	[sur:	[sur:	[sur:
noun: man	verb: give	noun: girl	noun: flower
fnc: give	arg: man girl flower	fnc: give	fnc: give
prn: 3	prn: 3	prn: 3	prn:3

These examples represent content at the context level because their **sur** (for *surface*) attributes have the value NIL (represented by empty space). If the proplets were to rep-

³Autonomy from the metalanguage, cf. FoCL'99, p. 64, 382.

⁴The term proplet was introduced in Hausser 1996 and coined in analogy to “droplet.” Proplets are so-called because they are the elementary items constituting a **proposition**.

resent content at the language level, the **sur** attributes would have non-NIL values, for example [sur: träumt].

The remaining attributes of the simplified proplets in 2.1 are interpreted as follows: The second attribute, **noun** or **verb**, is called the *core* attribute of a proplet, specifies the part of speech, and takes a basic meaning, e.g., a concept, as its value. The third attribute, **fnc** or **arg**, is called a *continuation* attribute and specifies the proposition's functor-argument structure. The fourth attribute, **prn**, is called a *book-keeping* attribute and takes a proposition number as its value; proplets belonging to the same proposition have the same proposition number.

As sets, proplets are unordered, and may be stored and retrieved according to the needs of one's database. Nevertheless, proplets code the traditional semantic relations of functor-argument structure and coordination, at the context level as well as the language level.

The method of coding semantic relations is called *symbolic bidirectional pointering*. It is symbolic because the relation between, e.g., a functor like **dream** and an argument like **girl** is represented by symbols (in the sense of computer science) serving as values. It is bidirectional because any functor specifies its argument(s) in its **arg** slot and any argument specifies its functor in its **fnc** slot, and similarly for modifiers and modifieds.

This method is suitable also for coordination,⁵ as illustrated by the following example:

2.2. Representing Julia sleeps. John dreams. Susanne sings.

$\left[\begin{array}{l} \text{sur:} \\ \text{noun: Julia} \\ \text{fnc: sleep} \\ \text{prn: 4} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{verb: sleep} \\ \text{arg: Julia} \\ \text{pc:} \\ \text{nc: 5 dream} \\ \text{prn: 4} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{noun: John} \\ \text{fnc: dream} \\ \text{prn: 5} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{verb: dream} \\ \text{arg: John} \\ \text{pc: 4 sleep} \\ \text{nc: 6 sing} \\ \text{prn: 5} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{noun: Susanne} \\ \text{fnc: sing} \\ \text{prn: 6} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{verb: sing} \\ \text{arg: Susanne} \\ \text{pc: 5 dream} \\ \text{nc:} \\ \text{prn: 6} \end{array} \right]$
---	---	--	---	--	--

The coordination is coded in the **pc** (for previous conjunct) and the **nc** (for next conjunct) slot of the verb of the elementary propositions, using a proposition number, e.g., **5**, and a verb, e.g., **dream**, as values.⁶ Note that the initial verb proplet of this extrapositional coordination has an empty **pc** slot, while the last verb proplet has an empty **nc** slot.⁷

3. Center Fragments for Different Word Orders

The semantic analysis of natural language meaning has two basic aspects, (i) lexical semantics and (ii) compositional semantics. In Database Semantics, the aspect of lexi-

⁵Just as the basic notion of functor-argument structure has been modeled in Predicate Calculus, the basic notion of coordination has been modeled in Propositional Calculus. However, while Predicate Calculus and Propositional Calculus are sign-oriented, metalanguage-based, and truth-conditional, Database Semantics is agent-oriented, procedural, and based on bidirectional pointering.

⁶For a detailed analysis of intra- and extrapositional coordination in English, including gapping, see NLC'06, Chapters 8 and 9.

⁷This is similar to the linear data structure of a linked list.

cal semantics is treated in terms of the core value of the proplets, while the aspect of compositional semantics is treated in terms of the continuation values. Accordingly, the three propositions coordinated in Example 2.2 differ in their lexical semantics, but their respective intrapositional compositional semantics are equivalent.

A center fragment focuses on the compositional aspects of natural language semantics, and may be characterized as follows:

3.1. The Seven Criteria of a Well-Defined Center Fragment

1. Semantic relations

A center fragment handles the primary semantic relations of natural language, namely functor-argument structure and coordination. These are assumed to be universal in the sense that they may be found in all natural languages.

2. Data coverage

A center fragment handles the primary semantic relations in principle, but in their simplest form. Therefore a center fragment is purposely limited to functor-argument structure at the level of *elementary* nouns, verbs, (e.g., Example 2.1) and adjectives, and to coordination at the *clausal* level (e.g., Example 2.2).

3. Functional standard

The functional standard of a center fragment is that of Database Semantics, i.e., modeling the mechanism of natural language communication in the speaker mode (language production) and the hearer mode (language interpretation).

4. Formal standard

A center fragment must be specified as an explicit formal definition suitable as the declarative specification for computational implementations.

5. Equivalence

The center fragments of different natural languages are called equivalent if they handle the same grammatical relations at the level of proplets. Thus, equivalent center fragments map a given set of proplets into equivalent surfaces of different languages (speaker mode) and equivalent surfaces of different languages into the same – or rather equivalent– sets of proplets (hearer mode).

6. Upscaling

There are two systematic ways of upscaling, namely grammar-based and frequency-based. Grammar-based upscaling consists in complementing the elementary nouns, verbs, and adjectives with their phrasal and clausal counterparts, and complementing clausal coordination with the coordination of elementary and phrasal nouns, verbs, and adjectives. Frequency-based upscaling consists in ordering the constructions found in a corpus and integrating those not yet handled, beginning with the most frequent.

7. Verification

A center fragment of a natural language and each subsequent step of upscaling must be verified computationally by implementing the system as a running program and testing it automatically on the data selected. Thereby the functional (cf. 3) and formal (cf. 4) standards of Database Semantics must be maintained in full.

Based on the representation of one-place, two-place, and three-place propositions in 2.1, let us define four equivalent center fragments for languages with different word orders.⁸ The following schema shows the word order variations of English, German, Korean, and Russian:

3.2. Schematic Comparison of Word Orders

	V-first	V-second	V-third	V-last
one-place	VS	SV	English	(SV)
two-place	VSO	SVO	-German	SOV
	VOS	OVS	Korean	OSV
three-place	VSID	SVID	SIVD	SIDV
	VSDI	SVDI	SDVI	SDIV
	VISD	IVSD	ISVD	ISDV
	VIDS	IVDS	IDVS	IDSV
	VDSI	DVSI	DSVI	DSIV
	VDIS	DVIS	DIVS	DISV

S stands for Subject or nominative, V for Verb, O for Object,⁹ D for Direct object or accusative, and I for Indirect object or dative. The permutations are ordered (i) according to the places of the verb (one-place in line 1, two-place in lines 2 and 3, and three-place in lines 4–9), and (ii) according to the position of the verb in the sentence (initial position in column 1, second position in column 2, third position in column 3, and last position in column 4).¹⁰ Due to different word order possibilities, the three propositions of 2.1 have a total of three surfaces¹¹ in English, of nine in German and Korean, and of thirty-two in Russian.

The challenge presented by the four different center fragments is to provide the correct semantic relations for the different word orders, using a strictly time-linear derivation order in the hearer mode. Consider the following example of an SVO derivation (e.g., English):

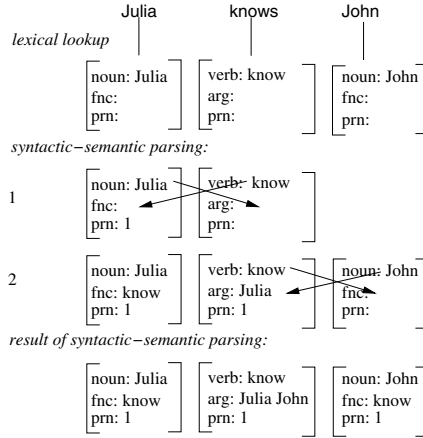
⁸For reasons of space, elementary adjectives and extrapositional coordination are omitted, and the formal center fragments are specified in the hearer mode only. In NLC'06, Chapters 11–14, complete fragments with LA-hear, LA-think, and LA-speak grammars are explicitly defined as declarative specifications which have been verified by a concomitant implementation in Java (Kycia 2004).

⁹The terminology of S, V, and O follows Greenberg 1963.

¹⁰Note that SV occurs twice in this table, once as verb-second (English and German) and once as verb-last (Korean and Russian).

¹¹For better comparison of the four center fragments, additional surfaces such as *The girl was seen by the man* (passive) or *the man gave a flower to the girl* (prepositional dative) are disregarded here.

3.3. Time-Linear Hearer-Mode Analysis of an SVO Surface

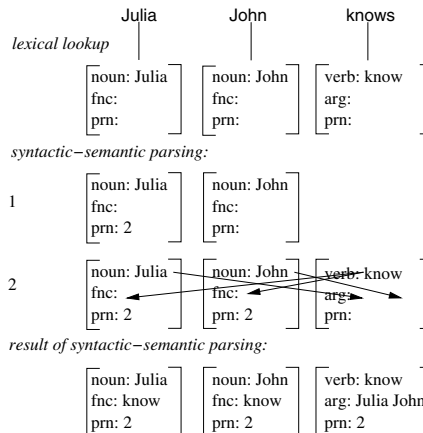


The derivation begins with an incremental lexical lookup, which relates each incoming surface to a corresponding proplet. These lexical proplets are called *isolated* proplets because their continuation attributes have no values yet. The isolated proplets are turned into *connected* proplets by a variant of LA-grammar, called LA-hear.

The task of an LA-hear grammar is to provide the most direct buildup of semantic relations possible in a strictly time-linear derivation order. This is done mainly by copying values between proplets. For example, in line 1 the core value of the *Julia* proplet is copied into the *arg* slot of the *know* proplet, and the core value of the *know* proplet is copied into the *fnc* slot of the *Julia* proplet (symbolic bidirectional pointering), and accordingly in line 2. The result is a content coded as a set of proplets ready to be sorted into the database.

While the SVO order permits immediate cross-copying between the verb and its arguments, this is not possible in the SOV order (e.g., Korean), as shown by the following example:

3.4. Time-Linear Hearer-Mode Analysis of an SOV Surface



In line 1, the first and the second noun cannot be related in terms of functor-argument structure (no verb yet). Such adding of a lexical proplet without building a semantic relation to any proplet(s) in the sentence start is called *suspension* (cf. NLC'06, Section 7.6). After a suspension, all the necessary cross-copying takes place as soon as possible and at once, as shown in line 2. The result is a content equivalent to the one derived in Example 3.3.

4. Fixed Noun Order, Verb Second: English

Having specified the language phenomena to be handled by our four center fragments intuitively, we turn now to defining formally explicit LA-hear grammars for them. An LA-hear grammar is a symbolic system which consists of (i) word form recognition, (ii) a variable definition, and (iii) syntactic-semantic rules. It is embedded into a language-independent software machine called the *motor*, currently implemented in Java.

In the hearer mode, the motor takes two kinds of language-dependent input, namely (a) an LA-hear grammar (prior to run time) and (b) unanalyzed surfaces of the language in question (during run time). The output of the motor are proplets which represent the content¹² coded by the input surfaces by means of symbolic bidirectional pointer-*ing*.

The linguistic analysis of an LA-hear grammar differs from the sign-oriented approaches of Nativism and Modeltheoretic Semantics in three fundamental ways:

4.1. Characteristic Properties of LA-Hear Grammars

1. Rules have an external interface based on pattern matching

The rules of LA-grammar in general and LA-hear in particular analyze input based on pattern matching, whereas the rules of the sign-oriented approaches are typically based on possible substitutions without any external interface.¹³

2. No separation of syntactic composition and semantic interpretation

LA-hear treats the syntactic composition and the semantic interpretation simultaneously, whereas the sign-oriented approaches typically handle the syntactic composition first and then add a separate semantic interpretation.

3. Strictly time-linear derivation order

LA-hear integrates the syntactic-semantic composition into a strictly time-linear derivation order which computes possible continuations, whereas the sign-oriented approaches combine the words and phrases according to what belongs together semantically, resulting in irregular hierarchies called constituent structure.

As a reference grammar for comparing different kinds of time-linear syntactic-semantic compositions in the following sections, let us define an LA-hear grammar for

¹²I.e., the compositional semantics (grammatical relations) of the input expressions.

¹³This applies specifically to the rewrite rules of context-free phrase structure grammar. Though Chomsky's transformations (cf. FoCL'99, Example 8.5.3) have an interface based on matching, it is only an *internal* interface (i.e., transformations modify phrase structure trees rather than taking external language input). The rules of Categorical Grammar (cf. FoCL'99, Sections 7.5, 7.5) have external interfaces, but they compute possible substitutions rather than possible continuation and are therefore not time-linear.

the center fragment of English, represented by the sentences analyzed in Example 2.1. To simplify matters, the word form recognition system for these sentences is (i) defined as a full-form lexicon and (ii) treats the noun phrases *the_girl*, *the_man* and *a_flower* as elementary proplets.¹⁴ To illustrate the handling of grammatical agreement, the word form recognition also handles the plural *the_girls* and the unmarked verb form *dream*:

4.2. Word Form Recognition System of LAH.English.1

$\left[\begin{array}{l} \text{sur: the_girl} \\ \text{noun: girl} \\ \text{cat: snp} \\ \text{fnc:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: the_girls} \\ \text{noun: girl} \\ \text{cat: pnp} \\ \text{fnc:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: the_man} \\ \text{noun: man} \\ \text{cat: snp} \\ \text{fnc:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: a_flower} \\ \text{noun: flower} \\ \text{cat: snp} \\ \text{fnc:} \\ \text{prn:} \end{array} \right]$
$\left[\begin{array}{l} \text{sur: dreams} \\ \text{verb: dream} \\ \text{cat: ns3' v} \\ \text{arg:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: dream} \\ \text{verb: dream} \\ \text{cat: n-s3' v} \\ \text{arg:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: sees} \\ \text{verb: see} \\ \text{cat: ns3' a' v} \\ \text{arg:} \\ \text{prn:} \end{array} \right]$	$\left[\begin{array}{l} \text{sur: gives} \\ \text{verb: give} \\ \text{cat: ns3' d' a' v} \\ \text{arg:} \\ \text{prn:} \end{array} \right]$

A system of automatic word form recognition like 4.2, based on lexical lookup from a full-form lexicon, is technically very simple: it consists of matching an incoming unanalyzed surface, e.g., *dreams*, with the *sur* value of the corresponding lexical full-form proplet.¹⁵

The second component of an LA-hear grammar, i.e., the variable definition, is needed for specifying the patterns which provide the LA-grammar rules with an external interface. In order for a rule pattern and a language proplet to successfully match, (i) the attributes of the rule pattern must be a subset of the attributes of the language proplet and (ii) the values of the rule pattern must be compatible with the corresponding values of the language proplet.

The values of a rule pattern consist of variables and constants. Because of the variables, a rule pattern (type) can match any number of language proplets (token) of the same kind. The compatibility of corresponding values at the rule and the language level is defined as follows: any constant at the rule level must be matched by the same constant at the language level, and any variable at the rule level must be matched by a constant at the language level which is in the *restriction set* of the variable.

Variables with general restrictions are written as lower case Greek letters like α , β , γ , which match any string of characters, or as X, Y, Z, which match any sequence of zero, one, two, or three constants. Variables with specific restrictions are defined in the variable definition:

4.3. Variable Definition of LAH.English.1

1. Variable restrictions:

$\text{NP} \in \{\text{snp}, \text{pnp}\}$

$\text{NP}' \in \{\text{ns3'}, \text{n-s3'}, \text{d'}, \text{a'}\}$

¹⁴For the proper incremental treatment see NLC'06, Chapter 13.

¹⁵For an overview of different methods of automatic word form recognition see FoCL'99, Chapters 13–15.

2. Variable constraints (agreement conditions):

If $NP \in \{snp\}$, then $NP' \in \{ns3', d', a'\}$.

If $NP \in \{pnp\}$, then $NP' \in \{n-s3', d', a'\}$.

The variable **NP** (for noun phrase) is restricted to the category segments **snp** (for singular noun phrase) and **pnp** (for plural noun phrase), which represent nominal valency *fillers*. The variable **NP'** is restricted to the category segments **ns3'** (for nominative singular 3rd person, e.g., **dreams**), **n-s3'** (for nominative minus singular 3rd person, e.g., **dream**), **d'** (for dative), and **a'** (for accusative), which represent nominal valency *positions* in the category of a verb.

According to the constraint, any rule containing the variables **NP** as well as **NP'** will only match the input at the language level if the variable **NP** is bound either (i) to the constant **snp** and the variable **NP'** is bound to a constant **ns3'**, **d'**, or **a'**, or (ii) **NP** is bound to the constant **pnp** and **NP'** is bound to a constant **n-s3'**, **d'**, or **a'**. In this way, agreement violations as in **The girl dream* or **The girls dreams* will not be accepted. As shown in FoCL'99, 17.4.1, and NLC'06, 13.2.1, the variable definition provides a simple and effective method for handling grammatical agreement in all its forms, without cluttering up the rule system.

The rule system of an LA-hear grammar defines (i) a set of start states ST_S , (ii) a set of rules, and (iii) a set of final states ST_F . Consider the following example:

4.4. Rule System of LAH.English.1

$ST_S =_{def} \{ ([cat: X] \{ 1 N+V \}) \}$

N+V $\{ 2 V+N \}$

$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix}$	$\begin{bmatrix} \text{verb: } \beta \\ \text{cat: NP'} \ Y \ v \\ \text{arg:} \end{bmatrix}$	<p>delete NP' nw.cat acopy α nw.arg ecopy β ss.fnc copy_{ss} copy_{nw}</p>
---	---	---

V+N $\{ 3 V+N \}$

$\begin{bmatrix} \text{verb: } \alpha \\ \text{cat: NP'} \ Y \ v \\ \text{arg:} \end{bmatrix}$	$\begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix}$	<p>delete NP' ss.cat acopy β ss.arg ecopy α nw.fnc copy_{ss} copy_{nw}</p>
--	--	---

$ST_F =_{def} \{ ([cat: v] \text{rp}_{N+V}), ([cat: v] \text{rp}_{V+N}) \}$

A start state specifies (i) a pattern for a possible first word of a sentence or text and (ii) a rule package listing all the rules which may apply to the initial word.¹⁶ A rule consists of (i) a rule name, (ii) a rule package, (iii) a pattern for the sentence start, (iv) a pattern for the next word, and (v) a set of operations.¹⁷ A final state specifies a pattern characterizing a complete sentence and the rule package of a sentence-completing rule.¹⁸

¹⁶Here, the pattern $[cat: X]$ will accept any first word. The rule package, $\{ 1 N+V \}$, contains only one rule and ensures that any derivation will begin with an application of the rule **N+V** (for noun plus verb).

¹⁷For example, the first rule of the above LA-hear grammar has (i) the name **N+V**, (ii) the rule package $\{ 2 V+N \}$, (iii) a sentence start pattern for a noun proplet, (iv) a next word pattern for a verb proplet, and (v) four operations.

¹⁸In the above example, there are two final states, one ending with an application of the rule **N+V** (in an intransitive sentence), the other ending in an application of the rule **V+N** (in a transitive sentence). The pattern $[cat: v]$ represents a verb proplet with no uncanceled valency positions.

As an illustration of a rule application, consider the following example. It shows the rule N+V defined in 4.4 applying to the proplets *man* and *see* defined in 4.2:

4.5. Example of an LA-Hear Rule Application

	N+V	{ 2 V+N }	
<i>rule level</i>	$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix}$	$\begin{bmatrix} \text{verb: } \beta \\ \text{cat: NP'} \ Y \ v \\ \text{arg:} \end{bmatrix}$	delete NP' nw.cat acopy α nw.arg ecopy β ss.fnc copy _{ss} copy _{nw}
<i>language level</i>	$\begin{bmatrix} \text{sur: the_man} \\ \text{noun: man} \\ \text{cat: snp} \\ \text{fnc:} \\ \text{prn: 7} \end{bmatrix}$	$\begin{bmatrix} \text{sur: sees} \\ \text{verb: see} \\ \text{cat: ns3'} \ a' \ v \\ \text{arg:} \\ \text{prn:} \end{bmatrix}$	

Matching between the rule level and the language level is successful because the attributes of the rule patterns are subsets of the attributes of the corresponding language proplets. Furthermore, the restrictions of the variables NP, NP', and Y as well as the constraint defined in 4.3 are satisfied.

During matching, the variables α and β are vertically bound¹⁹ to the values *man* and *see*, respectively, and the variables NP, NP' and Y are vertically bound to the values *snp*, *ns3'*, and *a'*, respectively. The vertical binding of rule level variables to language level values is the precondition for executing the operations at the language level. The assignment to a variable (scope) holds within a rule application.

The operation **delete** NP' nw.cat deletes the value assigned to the variable NP', i.e., *sn3'*, in the *cat* attribute of the next word proplet. The operation **acopy** α nw.arg adds the value assigned to the variable α , i.e., *man*, to the *arg* slot of the next word proplet, while the operation **ecopy** β ss.fnc copies the value assigned to the variable β , i.e., *see*, to the empty *fnc* slot of the sentence start. The operations **copy_{ss}** **copy_{nw}** retain the proplets of the sentence start and the next word proplet in the output.

The successful rule application shown in Example 4.5 has the following result:

4.6. Result of the LA-Hear Rule Application

$\begin{bmatrix} \text{sur: the_man} \\ \text{noun: man} \\ \text{cat: snp} \\ \text{fnc: see} \\ \text{prn: 7} \end{bmatrix}$	$\begin{bmatrix} \text{sur: sees} \\ \text{verb: see} \\ \text{cat: a'} \ v \\ \text{arg: man} \\ \text{prn: 7} \end{bmatrix}$
---	--

In addition to the execution of the operations, the control structure of the motor provides the *prn* attribute of the next word with a value, here 7. Next, the rule V+N contained in the rule package of N+V is applied to the verb proplet *see* in 4.6 and to the lexical proplet *girl* defined in 4.2. The overall derivation corresponds to 3.3.

¹⁹The vertical binding of variables in Database Semantics is in contradistinction to the horizontal binding of variables by quantifiers in Predicate Calculus (cf. NLC'06, Section 5.3).

5. Free Noun Order, Verb Second: German

Because English is a fixed word order language with the finite verb in post-nominative position in declarative sentences, each kind of functor-argument structure in Example 2.1 has only one corresponding surface. The nouns are morphologically unmarked for case,²⁰ and get their syntactic-semantic case role assigned by the valency position they fill in the verb.

German, in contrast, is a language with a comparatively free word order: the arguments (nouns) may occur in any order, while the finite verb must be in second position in declarative main clauses. This results in three times as many surfaces as in English:

5.1. The 9 Basic Functor-Argument Surfaces of German

- one-place verb: 1 surface
man_*nom* dream N+V
- two-place verb: 2 surfaces
man_*nom* see girl_*acc* N+V, V+N
girl_*acc* see man_*nom*
- three-place verb: 6 surfaces
man_*nom* give girl_*dat* flower_*acc* N+V, V+N, V+N
man_*nom* give flower_*acc* girl_*dat*
girl_*dat* give man_*nom* flower_*acc*
girl_*dat* give flower_*acc* man_*nom*
flower_*acc* give girl_*dat* man_*nom*
flower_*acc* give man_*nom* girl_*dat*

The examples are ordered into blocks, each followed by the associated rule sequence.

The free order of nouns in German is supported by morphological case marking. This is why the schematic full-form lexicon of English in Example 4.2 has only one entry for each singular noun, while the nouns in the corresponding full-form lexicon of German each have three entries (cf. 5.2 below), assuming unambiguous morphological case marking.

Besides German, Korean and Russian also have morphologically case-marked nouns, though each by different means: German by means of determiners and inflectional endings (though highly defective), Korean by agglutinative endings (cf. Lee 2004), and Russian by inflectional endings alone. Rather than defining the following word form recognition system only for German, let us define it for case marking languages in general.

Limited to the word forms needed for the respective center fragments of German, Korean, and Russian (cf. 3.2), this generic word form recognition system for case mark-

²⁰With the exception of the personal pronouns, which are morphologically distinguished between nominative and oblique case, e.g., *she* vs. *her*. For a detailed discussion see FoCL'99, Section 17.2.

ing languages handles three nouns unmarked for number, each in the nominative, dative, and accusative, and three verbs with valencies for one, two, and three arguments:

5.2. Word Form Recognition System for Case-Marked Nouns

$\begin{bmatrix} \text{sur: man_nom} \\ \text{noun: man} \\ \text{cat: n} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: man_dat} \\ \text{noun: man} \\ \text{cat: d} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: man_acc} \\ \text{noun: man} \\ \text{cat: a} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$
$\begin{bmatrix} \text{sur: girl_nom} \\ \text{noun: girl} \\ \text{cat: n} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: girl_dat} \\ \text{noun: girl} \\ \text{cat: d} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: girl_acc} \\ \text{noun: girl} \\ \text{cat: a} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$
$\begin{bmatrix} \text{sur: flower_nom} \\ \text{noun: flower} \\ \text{cat: n} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: flower_dat} \\ \text{noun: flower} \\ \text{cat: d} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: flower_acc} \\ \text{noun: flower} \\ \text{cat: a} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$
$\begin{bmatrix} \text{sur: dream_n} \\ \text{verb: dream} \\ \text{cat: n' v} \\ \text{arg:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: see_n+a} \\ \text{verb: see} \\ \text{cat: n' a' v} \\ \text{arg:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: give_n+d+a} \\ \text{verb: give} \\ \text{cat: n' d' a' v} \\ \text{arg:} \\ \text{prn} \end{bmatrix}$

The following variable definition is also generic, applying to all three of the case-marking languages in question:

5.3. Variable Definition for Case-Marking Languages

1. Variable restriction:

$$\begin{aligned} \text{NP} &\in \{n, d, a\} \\ \text{NP}' &\in \{n', d', a'\} \end{aligned}$$

2. Variable constraint (agreement conditions)

$$\begin{aligned} \text{If } \text{NP} &\in \{n\}, \text{ then } \text{NP}' \in \{n'\}. \\ \text{If } \text{NP} &\in \{d\}, \text{ then } \text{NP}' \in \{d'\}. \\ \text{If } \text{NP} &\in \{a\}, \text{ then } \text{NP}' \in \{a'\}. \end{aligned}$$

According to the variable constraint, an n' valency position in the verb can only be canceled by an n noun, and similarly for the other cases. Thus a rule with the variables NP and NP' will not match any language proplets which would, for example, assign the value d to NP and the value a' to NP' .

Assuming the abstract word form analysis 5.2 and the variable definition 5.3, the rule system of LAH.German.1 is defined as follows:

5.4. Rule System of LAH.German.1

$ST_S =_{def} \{ ([cat: X] \{1 N+V\}) \}$

$N+V \quad \{2 V+N\}$

$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix}$	$\begin{bmatrix} \text{verb: } \beta \\ \text{cat: X NP' Y v} \\ \text{arg:} \end{bmatrix}$	<p>delete $\{NP'\}$ nw.cat acopy α nw.arg ecopy β ss.fnc copy_{ss} copy_{nw}</p>
---	---	---

$V+N \quad \{3 V+N\}$

$\begin{bmatrix} \text{verb: } \alpha \\ \text{cat: X NP' Y v} \\ \text{arg:} \end{bmatrix}$	$\begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix}$	<p>delete NP' ss.cat acopy β ss.arg ecopy α nw.fnc copy_{ss} copy_{nw}</p>
--	--	---

$ST_F =_{def} \{ ([cat: v] rp_{N+V}), ([cat: v] rp_{V+N}) \}$

Even though the rule system 4.4 for English handles three surfaces with a fixed noun order and the rule system 5.4 for German handles nine surfaces with a free noun order, the grammars are very similar. In fact, the only difference is in the respective *cat* values of the rules' verb patterns. In English, this pattern is $[cat: NP' Y v]$, whereas in German it is $[cat: X NP' Y v]$.

In other words, in the English pattern, the valency position NP' is not preceded by the variable X , while in German it is. Thus, given the proper word form analyses of nouns and the associated variable definitions, a noun in English always cancels the leftmost valency position in the verb, thereby obtaining its case role. In German, in contrast, any valency position of the verb may be canceled during a time-linear derivation as long as the filler is of a corresponding case.

6. Free Noun Order, Verb Final: Korean

Unlike English, but similar to German, Korean has a free order of nouns. In contrast to German, however, the verb is in final position. Thus, the surfaces of Korean coding the basic functor-argument structures corresponding to Example 5.1 are as follows:

6.1. The 9 Basic Functor-Argument Surfaces of Korean

- one-place verb: 1 pattern
man_{nom} dream N+V
- two-place verb: 2 patterns
man_{nom} girl_{acc} see N+N, N+V
girl_{acc} man_{nom} see
- three-place verb: 6 patterns
man_{nom} girl_{dat} flower_{acc} give N+N, N+N, N+V
man_{nom} flower_{acc} girl_{dat} give
girl_{dat} man_{nom} flower_{acc} give
girl_{dat} flower_{acc} man_{nom} give
flower_{acc} girl_{dat} man_{nom} give
flower_{acc} man_{nom} girl_{dat} give

These surfaces are analyzed by only two rules. However, while these rules are called N+V and V+N in English and German, they are called N+N and N+V in LAH.Korean.1.

Because Korean nouns are case-marked,²¹ the definition of LAH.Korean.1 may reuse the generic word form recognition 5.2 and the generic variable definition 5.3. The main difference between LAH.Korean.1 on the one hand, and LAH.English.1 and LAH.German.1 on the other, is in the rule system:

6.2. Rule System of LAH.Korean.1

$$\mathbf{ST}_S =_{def} \{ ([cat: X] \{ 1 \text{ N+N}, 2 \text{ N+V} \}) \}$$

$$\mathbf{N+N} \quad \{ 3 \text{ N+N}, 4 \text{ N+V} \}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \end{bmatrix} \quad \text{copy}_{ss} \text{ copy}_{nw}$$

$$\mathbf{N+V} \quad \{ \}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{noun: } \gamma \\ \text{cat: NP}_3 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{verb: } \delta \\ \text{cat: } \{ \text{NP}'_1 \text{ NP}'_2 \text{ NP}'_3 \} \text{ v} \\ \text{arg:} \end{bmatrix} \quad \begin{array}{l} \text{cancel } \{ \text{NP}'_1 \text{ NP}'_2 \text{ NP}'_3 \} \text{ nw.cat} \\ \text{acopy } \alpha \beta \gamma \text{ nw.arg} \\ \text{ecopy } \delta \text{ ss.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \\ \text{fnc:} \end{bmatrix} \quad \begin{bmatrix} \text{verb: } \gamma \\ \text{cat: } \{ \text{NP}'_1 \text{ NP}'_2 \} \text{ v} \\ \text{arg:} \end{bmatrix} \quad \begin{array}{l} \text{cancel } \{ \text{NP}'_1 \text{ NP}'_2 \} \text{ nw.cat} \\ \text{acopy } \alpha \beta \text{ nw.arg} \\ \text{ecopy } \gamma \text{ ss.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix} \quad \begin{bmatrix} \text{verb: } \beta \\ \text{cat: NP}' \text{ v} \\ \text{arg:} \end{bmatrix} \quad \begin{array}{l} \text{delete NP}' \text{ nw.cat} \\ \text{acopy } \alpha \text{ nw.arg} \\ \text{ecopy } \beta \text{ ss.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\mathbf{ST}_F =_{def} \{ ([cat: v] \text{ rp}_{N+V}) \}$$

One difference is in the finite state transition networks defined by the rules and rule packages of LAH.English.1 and LAH.German.1 (cf. 8.2) compared to that of LAH.Korean.1 (cf. 8.3): First, in LAH.Korean.1 there are two start states, one for intransitive sentences like *man sleep*, beginning with N+V, the other for transitive sentences like *man girl see* or *man girl flower give*, beginning with N+N. Second, the rule package of N+N contains two rules, N+N for adding another noun, as in *man girl* + *flower*, and N+V for adding the final verb, as in *man girl* + *see*. Third, the rule N+V has an empty rule package, indicating the end of the derivation.²²

The other difference is in the patterns and operations of the rules themselves: The rule N+N simply adds the next word to the sentence start, without any cross-copying (suspension, cf. 3.4). Minimal change is provided by the control structure of the mo-

²¹ See Choe et al. 2007 for a detailed analysis of Korean noun phrases within Database Semantics.

²² This, however, is only temporary and due to the tiny size of the current fragment. For example, as soon as clause-final modifiers, interpunctuation, or extrapositional coordination are added, the rule package of N+V will not be empty, but contain the associated rules.

tor, which adds the current proposition number to the *prn* slot of the next word proplet.

The rule N+V has three subclauses, one for three-place verbs, one for two-place verbs, and one for one-place verbs. These clauses are tried on the input in sequence: If the clause for a three-place verb does not match, the one for a two-place verb is tried; if the clause for a two-place clause does not match, the one for a one-place verb is tried. If this one fails as well, the application of the rule N+V fails as whole.

As in German, the noun fillers may cancel valency positions in any order. Unlike German, however, the canceling takes place all at once at the end. For this, the matching between the rule patterns and corresponding language proplets needs to be adapted in N+V: the set parentheses { } in the *cat* pattern of the verb in the first clause, i.e., {NP'₁ NP'₂ NP'₃} v, indicate that the surface order of the noun fillers NP₁, NP₂, and NP₃ may differ from the order of the corresponding valency positions in the category of the verb, and similarly for the second clause. Consider the following example:

6.3. N+V Matching Free Noun Order and Three-Place Verb

<i>rule patterns</i>	$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \\ \text{fnc:} \end{bmatrix}$	$\begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \\ \text{fnc:} \end{bmatrix}$	$\begin{bmatrix} \text{noun: } \gamma \\ \text{cat: NP}_3 \\ \text{fnc:} \end{bmatrix}$	$\begin{bmatrix} \text{verb: } \delta \\ \text{cat: \{NP}'_1 \text{ NP}'_2 \text{ NP}'_3\} v \\ \text{arg:} \end{bmatrix}$
<i>language proplets</i>	$\begin{bmatrix} \text{sur: flower_acc} \\ \text{noun: flower} \\ \text{cat: a} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: man_nom} \\ \text{noun: man} \\ \text{cat: n} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: girl_dat} \\ \text{noun: girl} \\ \text{cat: d} \\ \text{fnc:} \\ \text{prn} \end{bmatrix}$	$\begin{bmatrix} \text{sur: give_s3+d+a} \\ \text{verb: give} \\ \text{cat: n' d' a' v} \\ \text{arg:} \\ \text{prn} \end{bmatrix}$

Here, NP₁, NP₂, and NP₃ are bound to the values a, n, and d, respectively, thus restricting NP'₁, NP'₂, and NP'₃ to the values a', n', and d', respectively (variable constraint of 5.3). Nevertheless, the *cat* values of the verb pattern are compatible with the *cat* values n' d' a' v of the *give* proplet, due to the set parentheses in the verb pattern. The new operation **cancel** differs from **delete** in that it fills more than one valency position.

7. Completely Free Word Order: Russian

In Russian, not only the noun order is free, but also the position of the verb. Given the three propositions underlying the center fragments of English (3 surfaces), German (9 surfaces), and Korean (9 surfaces), this results in a total of 32 surfaces (cf. 3.2). They are analyzed by three rules, called N+V, N+N, and V+N.

Because Russian nouns are case-marked, the definition of LAH.Russian.1 may reuse the generic word form recognition system 5.2 and the generic variable definition 5.3 (like LAH.Korean.1). The LAH.Russian.1 rule system has a start state with a rule package containing N+N for verb third or fourth (last), N+V for verb second, and V+N for verb first:

7.1. Rule System of LAH.Russian.1

$$\mathbf{ST}_S =_{\text{def}} \{ ([\text{cat}: X] \{1 \text{ N+N}, 2 \text{ N+V}, 3 \text{ V+N}\}) \}$$

$$\mathbf{N+N} \quad \{4 \text{ N+N}, 5 \text{ N+V}\}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \end{bmatrix} \quad \text{copy}_{ss} \text{ copy}_{nw}$$

$$\mathbf{N+V} \quad \{6 \text{ V+N}\}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{noun: } \gamma \\ \text{cat: NP}_3 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{verb: } \delta \\ \text{cat: } \{ \text{NP}'_1 \text{ NP}'_2 \text{ NP}'_3 \} \text{ v} \\ \text{arg:} \end{bmatrix} \quad \begin{array}{l} \text{cancel } \{ \text{NP}'_1 \text{ NP}'_2 \text{ NP}'_3 \} \text{ nw.cat} \\ \text{acopy } \alpha \beta \gamma \text{ nw.arg} \\ \text{ecopy } \delta \text{ ss.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP}_1 \\ \text{fnc:} \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP}_2 \\ \text{fnc:} \end{bmatrix} \quad \begin{bmatrix} \text{verb: } \gamma \\ \text{cat: } \{ X \text{ NP}'_1 \text{ NP}'_2 \} \text{ v} \\ \text{arg:} \end{bmatrix} \quad \begin{array}{l} \text{cancel } \{ \text{NP}'_1 \text{ NP}'_2 \} \text{ nw.cat} \\ \text{acopy } \alpha \beta \text{ nw.arg} \\ \text{ecopy } \gamma \text{ ss.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\begin{bmatrix} \text{noun: } \alpha \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix} \quad \begin{bmatrix} \text{verb: } \beta \\ \text{cat: } \{ X \text{ NP}' \} \text{ v} \\ \text{arg:} \end{bmatrix} \quad \begin{array}{l} \text{cancel } \{ \text{NP}' \} \text{ nw.cat} \\ \text{acopy } \alpha \text{ nw.arg} \\ \text{ecopy } \beta \text{ ss.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\mathbf{V+N} \quad \{7 \text{ V+N}\}$$

$$\begin{bmatrix} \text{verb: } \alpha \\ \text{cat: } X \text{ NP}' \text{ Y v} \\ \text{arg:} \end{bmatrix} \begin{bmatrix} \text{noun: } \beta \\ \text{cat: NP} \\ \text{fnc:} \end{bmatrix} \quad \begin{array}{l} \text{cancel NP}' \text{ ss.cat} \\ \text{acopy } \beta \text{ ss.arg} \\ \text{ecopy } \alpha \text{ nw.fnc} \\ \text{copy}_{ss} \text{ copy}_{nw} \end{array}$$

$$\mathbf{ST}_F =_{\text{def}} \{ ([\text{cat}: v] \text{ rp}_{\text{N+V}}), ([\text{cat}: v] \text{ rp}_{\text{V+N}}) \}$$

The rule packages of LAH.Russian.1 call a total of 7 rules, compared to 4 in LAH.Korean.1 and 3 in LAH.German.1.

The rule N+N is the same as in LAH.Korean.1: it simply collects the nouns, relying on a later application of N+V to provide the bidirectional pointer for coding the required semantic relations. The rule V+N is the same as in LAH.German.1: it utilizes the adjacency of the verb and the noun to establish the relevant aspect of the functor-argument structure between the two. The rule N+V has three clauses as in Korean; also, the set parentheses in the verb pattern indicate that the order of the fillers may differ from the order of the valency positions in the verb. Unlike N+V for Korean, however, N+V for Russian has a non-empty rule package, containing the possible successor rule V+N; also, the second and third clause have the additional variable *X* in the *cat* attribute of the verb pattern, formally indicating that in Russian there may be a remainder of uncanceled valency positions.

This variable provides for an important difference between Korean and Russian: in Korean, the sentence-final application of N+V requires that all valencies are being canceled and all functor-argument structures are being completed at once, while in Russian N+V may also apply prefinally, with the verb in second or third position and one or two subsequent applications of V+N.

8. Comparing the Four LAH Center Fragments

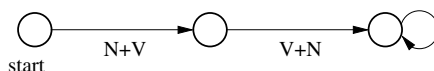
The four center fragments code the same content, but differ in whether nouns are case-marked (German, Korean, Russian) or not (English). They also differ in whether the order of nouns is fixed (English) or not (German, Korean, Russian). Finally, they differ in whether the position of the verb is fixed (English, German, Korean) or not (Russian):

8.1. Schematic Distinction Between the 4 Center Fragments

	case-marked	fixed noun order	fixed verb position
English	no	yes	yes
German	yes	no	yes
Korean	yes	no	yes
Russian	yes	no	no

In an LA-hear grammar, the distinction between nouns with and without case marking shows up in the automatic word form recognition and the variable definition. The distinction between fixed and variable noun order in English vs. German is treated minimally in terms of the absence vs. presence of the variable *X* in the *cat* pattern of the verb. Therefore, LAH.English.1 and LAH.German.1 have the same finite state transition network (FSN):

8.2. FSN Underlying LAH.English.1 and LAH.German.1

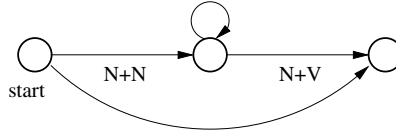


The network consists of states, represented by circles, and of rule applications, represented by arrows. Arrows going into a state correspond to the same rule called from different predecessor states, while arrows going out of a state (cf. 8.3, 8.4 below) correspond to the different rules of that state's rule package. With this in mind, it is sufficient for a complete characterization of the network to explicitly name only one of the arrows going into a state.

The rule *N+V* is different in English and German in that it assigns a nominative in English, whereas in German the case-marked noun cancels a corresponding valency position in the verb, and accordingly for the rule *V+N*. In sentences with a three-place verb, *V+N* applies to its own output, as indicated by the circle arrow. Given that an *V+N* application requires an uncanceled valency, the number of *V+N* repetitions is limited to one.

The rule systems of English and German on the one hand, and Korean on the other differ in that the former have two *simple cross-copying* rules (cf. 3.3), *N+V* and *V+N*, while Korean has a *suspension* rule *N+N* and a *multiple cross-copying* rule (cf. 3.4) *N+V* with three clauses. The finite state transition network underlying LAH.Korean.1 is as follows:

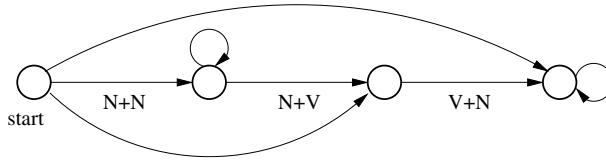
8.3. FSN Underlying LAH.Korean.1



As indicated by the circle arrow, N+N may apply to its own output, but only once and only if this is subsequently licensed by a three-place verb. If the verb is one-place, N+V is called by the start state.

The rule system of Russian combines the simple cross-copying rule V+N of English and German with the suspension rule N+N and the multiple cross-copying rule N+V of Korean, with concomitant changes of the rule packages:

8.4. FSN Underlying LAH.Russian.1



The respective rules N+V of LAH.Korean.1 and LAH.Russian.1 differ minimally in the absence vs. presence of the variable X in the *cat* pattern of the verb (compare 6.2 and 7.1), thus accounting for the fixed verb position in Korean and the free verb position in Russian.

We are now in a good position to establish the mathematical complexity of the four center fragments. This is important for the next steps, namely a systematic upscaling: If the center fragments can be shown to be of a low, i.e., linear, complexity, then each extension of the fragment should be designed in such a way that this degree of complexity is maintained.

The complexity of a grammar formalism – defined as the upper bound on the number of “primitive”²³ computational operations required in the analysis of arbitrary input – depends on the following two parameters:

8.5. Parameters of Complexity

- The *amount* of computation per rule application required in the worst case.
- The *number* of rule applications relative to the input length needed in the worst case.

These parameters are independent of each other and apply in principle to any rule-based grammar formalism (cf. FoCL9, Section 11.2).

In LA-grammar, the amount parameter depends on (i) the cost of matching the rule patterns with the input and (ii) the cost of applying the rule’s operations.²⁴ In basic propositions without modifiers, the maximal number of proplets is 4, namely one verb and at

²³This terminology follows Earley 1970.

²⁴These conditions differ from those of the earlier stage of LA-grammar, presented in NEWCAT’86 and analyzed in terms of complexity in TCS’92 and FoCL’99, Part II.

most three arguments, thus limiting the cost of matching with a small constant. Furthermore, the number of operations in each rule is finite. Therefore, as long as it holds for each available operation that the cost of its application is finite, the cost of applying the operations of any rule is finite.

Because each rule application in LA-grammar requires the consumption of a next word, the number parameter depends solely on the degree of ambiguity in a derivation. In the derivation of an unambiguous sentence, the maximal number of rule applications (including unsuccessful attempts) is $O \cdot (n - 1)$, whereby O is a constant representing the number of rules in the largest rule package of the grammar and n is the length of the sentence. In other words, the number of rule applications in an unambiguous LA-grammar grows linearly with the length of the input. It has been proven that the number of rule application grows only linearly even in ambiguous LA-grammars as long the ambiguities are non-recursive (cf. FoCL'99, 11.3.7, Theorem 3).

It follows that LAH.English.1, LAH.German.1, LAH.Korean.1, and LAH.Russian.1 are all of linear complexity. The cost of applying any of their rules is constant because the maximal number of proplets to be matched by the sentence start pattern is 3, the number of operations in each rule is finite, and the cost of applying them is finite as well (amount parameter). Furthermore, the LA-hear grammars in question are unambiguous because they don't have any rule package containing input-compatible²⁵ rules (number parameter).

9. Upscaling

After completing the center fragments of English, German, Korean, and Russian presented above,²⁶ they can handle a substantial number of grammatical constructions. Furthermore, the number of contents recognized and produced by a center fragment is infinite because there is no grammatical limit on the number of sentences in an extrapropositional coordination (text).

Nevertheless, the coverage of a center fragment is small compared to the whole of a natural language. This is in part because a natural language has nouns, verbs, and adjectives at the elementary, phrasal, and clausal level, while a center fragment is deliberately limited to the elementary level. Furthermore, natural language has extra- as well as intrapropositional coordination, while a center fragment is deliberately limited to extrapropositional coordination.

One direction to increase the coverage of a center fragment is by adding to the automatic word form recognition and production. Given (i) a suitable software, (ii) a tra-

²⁵Cf. FoCL'99, 11.3.2.

²⁶The extrapropositional coordination of basic propositions, illustrated in Example 2.2, requires two additional rules, called S+IP (sentence plus interpunctuation) and IP+START (interpunctuation plus sentence start), which are the same for all four fragments, and formally defined in NLC'06, Example 11.4.1.

Elementary modifiers, formally represented by proplets with the core attribute *adj*, for adjective, may be used adnominally or adverbially (cf. NLC'06, Section 6.3). While elementary propositions must have one verb and one, two, or three nouns (depending on the verb), adjectives are optional and may be stacked, with no grammatical limit on their number. Adding elementary adjectives in adnominal use requires two rules, DET+NN (determiner plus noun) and DET+ADN (determiner plus adnominal), which are formally defined in NLC'06, Example 13.2.4. A detailed hearer mode analysis of elementary and phrasal adjectives in adnominal and adverbial use may be found in NLC'06, Chapter 15.

ditional dictionary of the natural language existing online in public domain, and (iii) a well-trained computational linguist, a highly detailed word form recognition and production covering more than 90% of the word form types (!) in a sizeable corpus can be done in less than half a person year. The other direction to increase coverage is by extending the LA-hear, LA-think, and LA-speak grammars to additional constructions of the language in question. This may be done in two systematic ways, namely (i) grammar-based and (ii) frequency-based.

A grammar-based extension begins with a center fragment, which means that functor-argument structure is restricted to *elementary* nouns, verbs, and adjectives, while coordination is restricted to the sentence level. The first step of systematic grammar-based upscaling consists in adding *phrasal* nouns, verbs, and adjectives, and their *intrapropositional* coordination.²⁷ Then *clausal* nouns and adjectives are added (subordinate clauses):

9.1. Examples of Elementary, Phrasal, and Clausal Nouns

Elementary:	Julia saw Fido
Phrasal:	Julia saw a grumpy old dog
Clausal:	Julia saw that Fido barked

Also, the sentential mood of declarative is supplemented with other moods, such as interrogative and imperative, and their interpretation and production in dialogue. Such a grammar-based extension has been presented in NLC'06²⁸ for English.

A frequency-based extension begins with an almost complete grammar-based extension, including a largely complete automatic word form recognition of the language in question. The first step of a frequency-based extension is to apply the automatic word form recognition to a corpus. This allows to establish sequences of word form *categories* (rather than letter sequences representing surfaces), ordered according to frequency. Next, the grammar-based extension is tested automatically on this list of constructions (verification), starting with the most frequent ones. In this way, constructions not yet handled by the current system may be found and added to the existing DBS system.

Grammar-based extensions are important because they are based on semantic relations which are intuitively clear from the outset. Therefore, they may be implemented in the speaker and the hearer mode, and allow straightforward upscaling while maintaining the standard of functional completeness: an infinite set of constructions is composed recursively using only functor-argument structure and coordination, intrapropositionally and extrapropositionally. In addition, there is the secondary semantic relation of inference, which includes the handling of coreference.²⁹

Frequency-based extensions are necessary for achieving realistic coverage. This requires not only an extension of the grammar system to the new structures found in some corpus, but also a systematic development of the corpus itself. Such a corpus develop-

²⁷These include gapping constructions. Even though gapping constructions may be found very rarely in a corpus, their semantic relations are intuitively quite clear to the native speakers and hearers of a wide range of different languages.

²⁸NLC'06 contains over 100 detailed examples, analyzed in the hearer mode and the speaker mode.

²⁹Cf. NLC'06, Chapter 10.

ment should be based on a synchronic reference corpus structured into domains with subsequent annual monitor corpora. In this way, the coverage of a rule-based agent-oriented language analysis may be maintained continuously, extending its full functionality into the future. The possibilities of using such a system of Database Semantics for practical applications are enormous, and include all aspects of man-machine communication and natural language processing.

10. Conclusion

The starting point for the definition of equivalent center fragments are the most basic *contents* of compositional semantics. These are traditionally intrapropositional functor-argument structure (i.e., one-place, two-place, and three-place propositions) and extrapropositional coordination (concatenation of propositions as in a text). The scientific challenge is

1. to represent basic contents as a data structure suitable for storage and retrieval,
2. to decode the contents from surfaces with different word orders in the hearer mode,
3. to encode the contents into surfaces with different word orders in the speaker mode,
4. and to do the encoding and decoding in a strictly time-linear derivation order.

In this paper, different word orders are exemplified by the natural languages English, German, Korean, and Russian. For reasons of space, the definition of their equivalent center fragments is limited to the hearer mode and to elementary functor-argument structures without modifiers. The four center fragments are presented as formal definitions of partial DBS-systems, each consisting of an automatic word form recognition, a variable definition, and an LA-hearer grammar. These definitions serve as the *declarative specifications* for corresponding implementations (here in Java).

The theoretical as well as practical purpose of center fragments for natural languages is to ensure the success of long-term upscaling. Proof of concept is provided by NLC'06, in which a center fragment of English is systematically upscaled by extending functor-argument structure from elementary nouns, verbs, and adjectives to their phrasal and clausal counterparts, and by extending coordination from the clausal level to its phrasal and elementary counterparts, including gapping constructions. Equivalent upscaled fragments have been defined and computationally verified for German (Mehlhoff 2007) and Chinese (Hua Mei 2007).

References

- Austin, J.L. (1962) *How to Do Things with Words*. Oxford, England: Clarendon Press.
- Choe, Jae-Woong, and Hausser, R. (2007) "Treating quantifiers in Database Semantics," in M. Duži et al. (eds) *Information Modelling and Knowledge Bases XVIII*, Amsterdam: IOS Press.
- Chomsky, N. (1965) *Aspects of the Theory of Syntax*, MIT Press, Cambridge, Mass.
- Clark, H.H. (1996) *Using Language*. Cambridge, UK: Cambridge Univ. Press.
- Earley, J. (1970) "An efficient context-free parsing algorithm," *Commun. ACM* 13.2:94–102.
- Greenberg, J. (1963) "Some universals of grammar with particular reference to the order of meaningful elements," in J. Greenberg (ed.) *Universals of Language*. Cambridge, MA: MIT Press.

- Grice, P. (1965) "Utterer's meaning, sentence meaning, and word meaning," *Foundations of Language* 4:1–18.
- Hausser, R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283–308, Amsterdam: Elsevier (TCS'92).
- Hausser, R. (1996) "A database interpretation of natural language," *Korean Journal of Linguistics* 106(2), 29–55.
- Hausser, R. (1999) *Foundations of Computational Linguistics, 2nd ed. 2001*, Berlin Heidelberg New York: Springer (FoCL'99).
- Hausser, R. (2001) "Database Semantics for natural language," *Artificial Intelligence*, Vol. 130.1:27–74, Amsterdam: Elsevier (AIJ'01).
- Hausser, R. (2006) *A Computational Model of Natural Language Communication*, Berlin Heidelberg New York: Springer (NLC'06).
- Kučera, H. and W.N. Francis (1967) *Computational analysis of present-day English*, Brown U. Press, Providence, Rhode Island.
- Kycia, A. (2004) *A Java Implementating of the Speaker, Think, and Hearer Modes in Database Semantics*. MA-thesis, CLUE, Universität Erlangen–Nürnberg [in German].
- Mehlhoff, J. (2007) *An Implementation of a Center Fragment of German in Database Semantics*. MA-thesis, CLUE, Universität Erlangen–Nürnberg [in German].
- Mei, Hua (2007) *An Implementation of a Center Fragment of Chinese in Database Semantics*. MA-thesis, CLUE, Universität Erlangen–Nürnberg [in German].
- Montague, R. (1974) *Formal Philosophy*, Yale U. Press, New Haven.
- Roy, D. (2003) "Grounded spoken language acquisition: experiments in word learning," *IEEE Transactions on Multimedia* 5.2:197–209.
- Searle, J.R. (1969) *Speech Acts*. Cambridge, England: Cambridge Univ. Press.

Concepts and Ontologies

Marie DUŽÍ^a and Pavel MATERNA^b

^a*VSB-Technical University of Ostrava, Czech Republic*

marie.duzi@vsb.cz

^b*Institute of Philosophy, Czech Academy of Sciences, Czech Republic*

materna@lorien.site.cas.cz

Abstract. We introduce a new theory of concepts conceived as structured abstract entities. The theory is based on the key notion of Transparent Intensional Logic (TIL), known as TIL construction. The rich procedural semantics of TIL makes it possible to explicitly analyze all the semantically salient features of natural language expressions. We illustrate how to make use of TIL theory of concepts in distinguishing analytical and empirical concepts, and particular kinds of necessities. We also deal with a rigorous specification of requisite relations between intensions such as properties. Finally, ontology is characterised as a relatively stable part of the system that should play an integrating role. We show how to make use of TIL rich theory in specification of the content of ontologies in a multi-agent system.

Keywords. Ontology, concept, logical analysis of natural language, TIL, analytical vs. empirical concept, analytical vs. nomic necessity, intension, extension, hyper-intension

1. Introduction

As the title of this paper indicates we are going to talk about concepts and ontologies. One may ask, again? Aren't there numerous papers on this topic? Is there anything new to be presented? Even in the proceedings of the EJC conference you may find a couple of papers dealing with the topic, cf e.g., [6], [9], [10]. Yet we decided to open the topic again. There are several reasons. First, each of these papers deals with a particular problem, but none of them is a complex survey. The first of them is a paper introducing the theory of concepts as it has been worked out within the Transparent Intensional Logic (TIL). The second one deals with a special concept theory, namely that of concept lattices. And finally [10] concentrates rather on the *form*, in which ontology should be recorded and presented, than its *content* itself. Second, none of these papers deals with the practical problem what should be the content of ontology in case of a multi-agent system. This paper is inspired by the research and practice of a multi-agent system development conducted in our department.

The theory and practice of developing Multi-Agent System (MAS) has recently become a hot topic. A system of autonomous, intelligent yet resource-bounded agents who communicate and collaborate with each other in order to achieve their individual as well as collective goals is much more flexible than a classical centralized system. In particular in critical situations, where the traditional system is prone to chaotic behaviour or even collapse, particular agents of MAS can still survive and deal with the crisis. And since the agents live their individual lives, it might seem that there is no

need for a common ontology, schema, conceptual analysis, and other notions known from traditional system-development paradigms.

In our research we thus originally proposed the following decomposition of the MAS system development (see [8]):

- The *process-management* component that deals with the development of particular methods of agents' behaviour.
- The *knowledge-management* component that deals with knowledge representation, management and communication
- The *infrastructure component* that deals with space and time data, i.e., agents' environment
- All the components of the system are supported by a strong theoretical background of TIL. In this theoretical part we concentrated on the *logic* of agents' communication in a near-to-natural language and agents' reasoning.

In the first stage of development particular research groups worked almost separately. Thus we had 'process-agents', 'infrastructure-agents', 'communication-agents', 'knowledge-agents', and so like, but the system did not function as a whole. As soon as the integration into a real collaborative *system* was at need, it turned out that in the original architecture we neglected a component that was desperately needed; that of a *shared ontology*. Thus an ontology research group came spontaneously into being, the members of which designed ontology of a model, i.e., of a traffic system. But soon we saw, that the *form* in which the ontology is recorded is not the greatest problem; rather, it is the *content* that should be recorded in a particular ontology.

In general, ontology should contain a relatively stable part of the system in order to serve its integrating role. But what does it mean, a 'stable part'? Well, classically it is the result of a conceptual analysis of the system, i.e., of its functional and data component(s). This is perfectly true in case of a centralised system. But in a system of autonomous agents each agent may have its own vocabulary, terms, executive methods, knowledge and data. Particular agents can even speak their own languages. Yet, in order the agents can communicate (which they have to), learn from experience and collaborate, they have to share at least some part of a common *conceptual system*. No matter in which language particular concepts, methods and knowledge are encoded, the integral part of the system is its conceptual part.

This paper aims at two goals. First, we briefly recapitulate the basic notions of the theory of concepts, as partly presented in the previous papers, in particular those that turned out to be very important in the practical development of the MAS system ontology. We explicate what is meant by a concept and conceptual system, and how are they related to a particular language. Then we describe the methodological aspects of building up ontology content. Concluding, current stage of development of a computational variant of TIL, viz. the TIL-Script language is briefly described and future research is outlined.

2. Concepts: What are they?

Knowledge of the world is conveyed to us by language. Using expressions like 'the Moon', 'cat', 'to calculate', '(is) bigger than' we talk about extra-linguistic objects beyond the language: the Moon, the property of being a cat, the relation between an individual and some mathematical object, the relation between two material objects,

etc. There is, however, no intrinsic relation between the expressions as sequences of letters/sounds and the objects they speak about. Thus we have to assume that between an expression *E* of a natural language (a ‘NL expression’) and the object *E* is about (the ‘denotation of *E*’) there is some ideal entity (Frege called it ‘sense’, now we often use the term ‘meaning’) that is associated with *E* due to an anonymous linguistic convention and that makes it possible to identify the respective denotation (if any).

In general, we accept and adjust the Neo-Fregean conception of A. Church:

[a] name (i.e., an expression —our note) *denotes* or *names* its *denotation* and *expresses* its sense. [...] Of the sense we say that it *determines* the denotation, or *is a concept* of the denotation. [...] *Concepts* we think of as non-linguistic in character ...

We are even prepared to suppose the existence of concepts of things which have no name in any language in actual use. [...] The possibility must be allowed of concepts which are not concepts of any actual thing, and of names which express a sense but have no denotation. [...] To understand a language fully, we shall hold, requires knowing the senses of all names in the language, but not necessarily knowing which senses determine the same denotation, or even which senses determine denotations at all.

([3], pp. 6–7)

The term *concept* (or any equivalent in various languages) is frequently used but rarely well understood. In what follows we will try to explicate this term in such a way that its utility were obvious.

First of all, we do not deal with concepts understood as mental entities. Especially in cognitive sciences concepts are conceived and examined as mental entities, images or representations in the psychological sense. However, this conception is futile in the area of Information Technology for the following reasons:

1. Images in the psychological sense are subjective; thus there is no way to explicate the fact that particular NL users share common concepts as objective meanings of NL expressions;
2. In mathematics and other abstract disciplines we share concepts of abstract objects while there is no way to have an image of an abstract object;
3. Image is a concrete event dwelling here or there when somebody possesses it. It is a spatio-temporally localizable object whereas no concept can be spatio-temporally localized.

Hence we are going to treat *concepts* as *objective, non-mental* entities that can play the role of meanings of expressions, and since we need to use them in the information technology, our goal is to explicate concepts from the logical point of view. A bit of historical background first.

Plato's ideas were meant to answer the question “How does it come that we can know *universalia*, i.e., abstract universals?” They were eternal universals immune from sophistic relativism; being abstract they cannot evolve or change. (Similarly as concepts do not undergo a change; their seeming emergence is the emergence of a language: what is being changed is the link between an expression and its meaning, i.e., a concept).

Aristotle's concepts are from the contemporary viewpoint *definitions*: they create a new concept by composing simpler concepts (*genus proximum + differentia specifica*). Observe: In general they are therefore *structured*. The best exploitation of Aristotelian theory from the viewpoint of contemporary logic can be found in the work of R. Kauppi.¹

Similarly, *Bolzano's concepts* (*Begriffe*) are abstract extra-linguistics objects (see [2]). They are structured *ways of composing* particular simple concepts into complex ones.

Frege defined concepts as functions that associate objects with truth-values, and thus his concepts are characteristic functions of classes. He oscillated between two views that could be called in modern terminology 'function-in-extension' and 'function-in-intension', the former being a set-theoretical mapping, the latter a procedure computing the mapping. His conception is too narrow (only predicates are connected with concepts) and does not make it possible to capture the character of empirical concepts.

For *Church* concepts are potential meanings of meaningful expressions. One and the same concept is what any two synonymous expressions have in common. He defines synonymy of expressions as λ -convertibility; Church—as the founder of λ -calculi—is very close to the *procedural* theory of structured concepts (see [3]).

Tichý was presumably the first who explicitly declared that concepts are abstract *procedures* so that a logical theory of concepts should be based on the theory of *algorithms* (see [21], [22]). Later he founded Transparent Intensional Logic (TIL) where abstract procedures are explicated as *constructions* (see [23]).

Among the other authors declaring the procedural conception of concepts/meanings we find *Moschovakis*, whose (1994) paper has the symptomatic title "Sense and Denotation as Algorithm and Value" (see [19]).

Some other authors also recognize the necessity of conceiving meanings as *structured* entities. *Cresswell* in his (1985), see [5], introduced structured meanings as tuples. However, Cresswell's tuple theory actually does not meet the goal to render *structured* meanings. Thus, for instance, the meaning of the expression '*Seven minus five*' is modelled by Cresswell as the tuple $\langle M(''), M('7'), M('5') \rangle$ for *M* being the meaning of. It might seem that such a 'tuple-meaning' is structured, for we can see its particular parts which correspond to the parts of the expression. However, there are two problems here. First, these tuples are set-theoretical entities that do not present anything. Yet the above expression obviously presents (denotes) the number 2. The *way* of combining particular parts together is missing, i.e., the *instruction* of *applying* the function *minus* to the argument $\langle 7, 5 \rangle$. And it is of no avail to add the application of subtracting to the tuple to somehow yield the result 2, since the application would be merely another element of the tuple, not producing anything.² Second, using the 'tuple-meaning' theory, there is no way to define a function. For instance, the function of subtracting 5 (from a given argument) remains the same when being applied to 7, 8, or any other number. But while the procedure of declaring the function has such input/output gaps, the tuple does not. Removing the second element of the tuple $\langle M(''), M('7'), M('5') \rangle$ and replacing it with a 'gap-position' does not serve the goal because the result would be a "non-tuple". Tuples are set-theoretical objects, and all

¹ See [14]; Kauppi's theory is also well presented and explicated by J. Palomäki in [20].

² See [4], p. 51, for a recent statement of this objection

sets, unlike procedures, are algorithmically simple, have no such ‘input/output gaps’, and are flat mappings (characteristic functions).

Explication of the notion of *concept* is important also w.r.t. the goal of any rational communication, viz. intelligibility; it is not at all obvious that we understand each other even if we use the same language. A schematic example; being confronted with the following expressions,

‘*Natural numbers greater than 1 divisible just by 1 and themselves*’

‘*Natural numbers possessing just two factors*’,

would we say that we deal with two distinct concepts, or with just one concept expressed in two different ways?

Gödel, for example, voted for the latter option (see his [12]) whereas Bolzano (see [2]) would choose the former one, as well as Bealer (see [1]) and TIL do. Taking into account the fact that the notion of *concept* is used very frequently and in some contexts it is a very important (perhaps a key) notion, in particular when ‘ontologies’ are designed, we have to make our choice; otherwise some claims containing the term *concept* could be dubious or misunderstood. We hazard the risk of the former choice, similarly as Bolzano and Bealer did. There are many reasons for this decision, the most important of which is the sensitivity of agents’ attitudes to the *mode of presentation* (Frege’s sense) of the entity referred to by an expression. And we explicate this ‘mode’ as an instruction / procedure that yields the entity as its product. Moreover, even if there is no entity referred to by a (reasonable) expression *E*, like in case of ‘the greatest prime’, we do understand *E*; there is a concept expressed by *E*.

So how to explicate concepts from the *procedural* point of view? To this end we first need to introduce basic notions of TIL.

2.1 Transparent Intensional Logic (TIL)

Now we only briefly recapitulate; exact definitions can be found, e.g., in [6], [15], [17], [23], [24]. TIL *constructions* are uniquely assigned to expressions as their structured meanings. Intuitively, construction is a procedure (an instruction or a generalised algorithm), that consists of particular sub-instructions on how to proceed in order to obtain the output entity given some input entities. There are two kinds of constructions, atomic and compound. Atomic constructions (*Variables* and *Trivializations*) do not contain any other constituent but itself; they supply objects (of any type) on which compound constructions operate. *Variables* x, y, p, q, \dots , construct objects dependently on a valuation; they v -construct. *Trivialisation* 0X of an object X (of any type, even a construction) constructs simply X without the mediation of any other construction.³ *Compound* constructions, which consist of other constituents, are *Composition* and *Closure*. *Composition*, $[F A_1 \dots A_n]$, is the instruction to apply a function f (v -constructed by F) to an argument A (the tuple v -constructed by $A_1 \dots A_n$).⁴ Thus it v -constructs the value of f at A , if the function f is defined at A , otherwise the Composition is v -improper, i.e., it does not v -construct anything. *Closure*, $[\lambda x_1 \dots x_n X]$, is the instruction to v -construct a function by abstracting over variables x_1, \dots, x_n in the ordinary manner

³ In programming languages you may find a similar mechanism of a (fixed) pointer to an entity X and its dereference.

⁴ We treat functions as mappings, i.e., set-theoretical objects, unlike the *constructions* of functions.

of λ -calculi.⁵ Finally, higher-order constructions can be used twice over as constituents of composed constructions. This is achieved by a fifth construction called *Double Execution*, 2X , that behaves as follows: If X v -constructs a construction X' , and X' v -constructs an entity Y , then 2X v -constructs Y ; otherwise 2X is v -improper.

TIL constructions, as well as the entities they construct, all receive a type. The formal ontology of TIL is bi-dimensional; one dimension is made up of constructions, the other dimension encompasses non-constructions. On the ground level of the type-hierarchy, there are non-constructional entities unstructured from the algorithmic point of view belonging to a *type of order 1*. Given a so-called *epistemic (or 'objectual')* base of *atomic types* (\mathbf{o} -truth values, \mathbf{t} -individuals, $\mathbf{\tau}$ -time moments/real numbers, $\mathbf{\omega}$ -possible worlds), the induction rule for forming functions is applied: where $\alpha, \beta_1, \dots, \beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times \dots \times \beta_n$ to α , denoted $(\alpha \beta_1 \dots \beta_n)$, is a type of order 1 as well.⁶ Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted by $*_1$. This type $*_1$ together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, type $(\alpha \beta_1 \dots \beta_n)$, involving $*_1$ in their domain or range is a *type of order 2*. Constructions belonging to a type $*_2$ that identify entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*. And so on *ad infinitum*.

An object A of a type α is called an α -object, denoted A/α . That a construction C v -constructs an α -object is denoted $C \rightarrow_v \alpha$. Quantifiers, \forall^α (the general one) and \exists^α (the existential one), are of types $(\mathbf{o}(\mathbf{o}\alpha))$, i.e., sets of sets of α -objects.⁷ $[\forall^\alpha \lambda x A]$ v -constructs True iff $[\lambda x A]$ v -constructs the whole type α , otherwise False; $[\exists^\alpha \lambda x A]$ v -constructs True iff $[\lambda x A]$ v -constructs a non-empty subset of the type α , otherwise False. We write ' $\forall x A$ ', ' $\exists x A$ ' instead of ' $[\forall^\alpha \lambda x A]$ ', ' $[\exists^\alpha \lambda x A]$ ', respectively, when no confusion can arise. We use an infix notation without trivialisation when using constructions of truth-value functions of type $(\mathbf{o}\mathbf{o}\mathbf{o})$, i.e., \wedge (conjunction), \vee (disjunction), \supset (implication), \equiv (equivalence), and negation (\neg) of type $(\mathbf{o}\mathbf{o})$, and when using a construction of an identity.

$(\alpha\text{-})$ intensions are members of a type $(\alpha\omega)$, i.e., functions from possible worlds to an arbitrary type α . $(\alpha\text{-})$ extensions are members of the type α , where α is not equal to $(\beta\omega)$ for any β , i.e., extensions are not functions with the domain of possible worlds. Intensions are frequently functions of a type $((\alpha\tau)\omega)$, i.e., functions from possible worlds to *chronologies* of the type α (in symbols: $\alpha_{\tau\omega}$), where a chronology is a function of type $(\alpha\tau)$.

We will use variable w as v -constructing elements of type ω (possible worlds), and t as v -constructing elements of type τ (times). If $C \rightarrow \alpha_{\tau\omega}$ v -constructs an α -intension, the frequently used Composition of the form $[[Cw]t]$, the intensional descent of the α -intension, is abbreviated as $C_w t$.

Some important kinds of intensions are:

⁵ Comparison with programming languages might be helpful: λ -Closure corresponds to a declaration of a procedure F with formal parameters x_1, \dots, x_n ; Composition corresponds to calling the procedure F with actual values A_1, \dots, A_n of parameters.

⁶ TIL is an open-ended system. The above epistemic base $\{\mathbf{o}, \mathbf{t}, \mathbf{\tau}, \mathbf{\omega}\}$ was chosen, because it is apt for natural-language analysis, but the choice of base depends on the area to be analysed.

⁷ Collections, sets, classes of ' α -objects' are members of a type $(\mathbf{o}\alpha)$; TIL handles classes (subsets of a type) as characteristic functions. Similarly relations (-in-extension) are of type(s) $(\mathbf{o}\alpha_1 \dots \alpha_m)$.

Propositions, type $o_{\tau\omega}$. They are denoted by empirical (declarative) sentences.

Properties of members of a type α , or simply *α -properties*, type $(o\alpha)_{\tau\omega}$. General terms (some substantives, intransitive verbs) denote properties, mostly of individuals.

Relations-in-intension, type $(o\beta_1 \dots \beta_m)_{\tau\omega}$. For example transitive empirical verbs, also attitudinal verbs denote these relations.

α -roles, offices, type $\alpha_{\tau\omega}$, where $\alpha \neq (o\beta)$, frequently $\iota_{\tau\omega}$; often denoted by concatenation of a superlative and a noun (like ‘*the highest mountain*’).

Example 1: We are going to analyse the sentence “Charles is looking for a parking lot”. Our method of analysis consists of three steps:

- 1) *Type-theoretical analysis*, i.e., assigning types to the objects talked about by the analysed sentence. In our case we have:

Charles/ ι ; *Look_for*/($o(\iota o)_{\tau\omega}$) $_{\tau\omega}$ —the relation-in-intension of an individual to a property of individuals: the seeker wants to find an instance of the property;
Parking(Lot)/($o(\iota o)_{\tau\omega}$)—the property of individuals.

- 2) *Synthesis*, i.e., composing the constructions of the objects *ad* (1) in order to construct the proposition of type $o_{\tau\omega}$ denoted by the whole sentence. The sentence claims that the individual Charles has the ‘seeking-property’ of looking for a parking Lot. Thus we have to construct the individual Charles, the ‘seeking-property’, and then apply the latter to the former. Here is how:

- a) The atomic construction of the individual called Charles is simply ${}^0\text{Charles}$.
- b) The ‘seeking-property’ has to be constructed by Composing the relation-in-intension *Look_for* with a seeker $x \rightarrow \iota$ and the property *Parking*/($o(\iota o)_{\tau\omega}$) an instance of which is being sought. But the relation-in-intension cannot be applied directly to its arguments. It has to be extensionalized first: $[[{}^0\text{Look_for } w] \iota]$, abbreviated as ${}^0\text{Look_for}_{wt}$. Thus we have:
 $[{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]$, v -constructing a truth-value. Abstracting first from x by $\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]$ we obtain the class of individuals; abstracting from w and t we obtain the ‘seeking-property’:

$$\lambda w \lambda t [\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]].$$

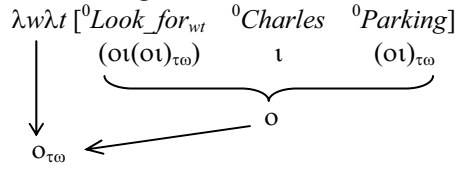
- c) Now we have to Compose the property constructed *ad* (b) with the individual constructed *ad* (a). The property has to be subjected to the intensional descent first, i.e., $[\lambda w \lambda t [\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]]]_{wt}$, and then Composed with the former;⁸ $[[\lambda w \lambda t [\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]]]_{wt} {}^0\text{Charles}]$.
- d) Since we are going to construct a proposition, i.e., an intension, we finally have to abstract from w, t :

$$\lambda w \lambda t [[\lambda w \lambda t [\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]]]_{wt} {}^0\text{Charles}].$$

This construction is the literal analysis of our sentence. It can still be β -reduced to the equivalent form:

$$\lambda w \lambda t [{}^0\text{Look_for}_{wt} {}^0\text{Charles} {}^0\text{Parking}].$$

⁸ For details on predication of properties and relations-in-intension of individuals, see [13].

3) *Type-Theoretical checking:*

The role of Trivialisation and empirical parameters $w \rightarrow \omega$, $t \rightarrow \tau$ in the communication between agents can be elucidated as follows. Each agent has to be equipped with a basic ontology, namely the set of simple concepts she is informed about. Thus the upper index ‘ 0 ’ serves as a marker of the simple concept that the agents should have in their ontology. If they do not, they have to learn them by asking the others. The lower index ‘ $_{wt}$ ’ can be understood as an instruction to execute an *empirical inquiry (search)* in order to obtain the actual current value of an intension, for instance by searching agent’s knowledge base or by asking the other agents, or even by means of agent’s sense perception.

2.2 Concepts as Closed Constructions

In the above example we mentioned the category of ‘*simple concepts*’; they are Trivialisations of the respective (non-constructive) entities.⁹ Indeed, the simplest meaning of the proper name ‘Charles’ is the Trivialisation of the individual called Charles: $\overset{0}{\text{Charles}}$. Similarly, the simplest meaning of ‘parking lot’ is the Trivialisation of the denoted property of individuals: $\overset{0}{\text{Parking}}$.

However, these concepts are not particularly informative. We might certainly refine, for instance, the meaning of ‘parking lot’ by *defining* it using some other well-known simple concepts, like ‘the place in the city where cars are allowed to park’. In the above example we defined the ‘seeking property’ as the relation-in-intension of an individual (seeker) to the property of being a parking lot by composing $\overset{0}{\text{Look_for}}$ and $\overset{0}{\text{Parking}}$: $\lambda w \lambda t [\lambda x [\overset{0}{\text{Look_for}}_{wt} x \overset{0}{\text{Parking}}]]$. On the other hand, the *open* constructions $\overset{0}{\text{Look_for}}_{wt}$, $[\overset{0}{\text{Look_for}}_{wt} x \overset{0}{\text{Parking}}]$, and $[\lambda x [\overset{0}{\text{Look_for}}_{wt} x \overset{0}{\text{Parking}}]]$ do not define anything; they just *v-construct* (waiting for the parameters of types ω , τ , ι , respectively).

Thus it is natural to define concepts as *closed* constructions. Closed constructions are assigned to expressions with a complete meaning¹⁰ as their meanings. They identify, or construct, the respective entities the expressions talk about. And they are algorithmically structured ways to those denoted entities. Moreover, unlike classical set-theoretical semantics, constructions are fine-grained enough to distinguish between meanings of non-synonymous, yet equivalent expressions that denote one and the same entity.

Consider, e.g., two different definitions of the set of primes: ‘natural numbers with just two factors’ vs. ‘natural numbers greater than one that are divisible only by one and the number itself’. We would hardly say that these two definitions are

⁹ For discussion on simple concepts and simple expressions, see [16].

¹⁰ i.e., expressions without indexicals. Expressions like ‘it is a parking place’ do not have a complete meaning, which does not mean, however, that they are meaningless: *open constructions* are assigned to them as their meaning.

synonymous though they denote one and the same set of numbers. Our TIL analyses reveal the respective (different) constructions/concepts:

Example 2 ‘natural numbers with just two factors’;

Types: $x, y \rightarrow \tau$; $\text{Nat}(\text{ural numbers})/(\text{o}\tau)$; $\text{Card}(\text{inality})/(\tau(\text{o}\tau))$ —cardinality of a class of numbers; $\text{Div}/(\text{o}\tau\tau)$ to be divisible by ($[^0\text{Div } x y]$ read ‘ x is divisible by y ’, applied to natural numbers only); $=/(\text{o}\tau\tau)$; $2/\tau$.

The following construction is a concept of prime numbers:

$$\lambda x [[^0\text{Nat } x] \wedge [[^0\text{Card } \lambda y [^0\text{Div } x y]] = ^0 2]].$$

Example 3 ‘natural numbers greater than one that are divisible only by one and the number itself’;

Types: $x, y \rightarrow \tau$; $\text{Nat}(\text{ural numbers})/(\text{o}\tau)$; $=, > / (\text{o}\tau\tau)$; $\forall/(\text{o}(\text{o}\tau))$ the universal quantifier; $\text{Div}/(\text{o}\tau\tau)$; $1/\tau$.

The following construction is another concept of prime numbers:

$$\lambda x [[^0\text{Nat } x] \wedge [^0 > x ^0 1] \wedge [^0 \forall \lambda y [[^0\text{Div } x y] \supset [[x = ^0 1] \vee [x = y]]]]].$$

The last point in this brief survey of TIL theory of concepts.¹¹ As mentioned above, it was as early as in 1968 when Tichý began to talk about *concepts* as *procedures*. We proposed closed TIL-constructions as good candidates to explicate concepts. They are fine-grained procedures corresponding in a near-to-isomorphic way to the syntactic structure of an expression to which the respective construction is assigned as its meaning. The question is, whether they are not *too* fine-grained. Consider, e.g., the above definition of primes ‘natural numbers with just two factors’ and its corresponding meaning: $\lambda x [[^0\text{Nat } x] \wedge [[^0\text{Card } \lambda y [^0\text{Div } x y]] = ^0 2]]$. From the conceptual point of view, it does not matter which particular λ -bound variables are used. We would surely say that the following constructions are the same concept of primes as the one above:

$$\lambda y [[^0\text{Nat } y] \wedge [[^0\text{Card } \lambda x [^0\text{Div } y x]] = ^0 2]],$$

$$\lambda z [[^0\text{Nat } z] \wedge [[^0\text{Card } \lambda y [^0\text{Div } z y]] = ^0 2]], \text{ etc.}$$

In natural language we cannot even express their difference, because they differ only in using λ -bound variables; in other words, they are α -equivalent.

Thus TIL concepts are conceived of as a special kind of constructions. A *concept* is represented by any closed construction (i.e., construction without free variables) that is an element of the class of *quasi-identical* constructions; quasi-identity on constructions is the relation of *procedural equivalence*. It is induced by α -equivalence and η -equivalence. To adduce an example, the following constructions represent one and the same concept:

$$^0\text{Parking}, \lambda w [^0\text{Parking } w], \lambda w \lambda t [[^0\text{Parking } w] t], \lambda w \lambda t \lambda x [[^0\text{Parking } w] t] x], \\ \lambda w \lambda t \lambda y [[^0\text{Parking } w] t] y], \lambda w \lambda t \lambda z [[^0\text{Parking } w] t] z], \dots \text{ ad infinitum.}$$

(The first four constructions are η -equivalent, the rest is the beginning of an infinite sequence of α -equivalent constructions.) Note that particular elements of this class are not identical, but they differ in a way irrelevant from the conceptual point of view; in natural language all of them are expressed by the term ‘*parking*’.

¹¹ For details see, e.g., [15], [17].

Concluding this brief survey, and following the intuition according to which a concept should unambiguously lead to an object (if any), we can state: *The meaning of any expression that does not contain indexicals or anaphoric references is a concept.*¹²

2.3 Analytical and Empirical Concepts

Now we are going to examine particular categories of concepts. The concepts represented by Examples 2 and 3 which identify extensions are *analytical concepts*: they are used to determine mathematical objects and their role consists simply in constructing the respective object. They often construct infinite functions and are satisfactory tools for their specification. Moreover, if the respective procedure is an effective one — i.e., the so-constructed function is *recursive* — then we can calculate the value of such a function at any argument.

Empirical concepts identify intensions. To adduce an example of an *empirical concept*, consider the analysis presented in Example 1:

$$\lambda w \lambda t [[\lambda w \lambda t [\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]]]]_{wt} {}^0\text{Charles}].$$

It is a concept of a proposition, the intension of type $\alpha_{\tau\omega}$. True, some logicians wouldn't call it a concept, because they conceive concepts as universalia, i.e., usually concepts of individual properties. But in accordance with Church's conception we conceive even the constructions of propositions, i.e., meanings of affirmative sentences, as concepts. Particular closed subconstructions of the above concept are again concepts, e.g., the concept of the 'seeking-property' is:

$$[\lambda w \lambda t [\lambda x [{}^0\text{Look_for}_{wt} x {}^0\text{Parking}]]].$$

Here is another example of a concept of a property:

Example 4 'concrete buildings older than 15 years';

Types: $x \rightarrow \iota$; *Concrete*/ $((\alpha\iota)_{\tau\omega}(\alpha\iota)_{\tau\omega})$ —modifier: applied to a property returns another property; *Building*/ $(\alpha\iota)_{\tau\omega}$ —a property of individuals; *Age_of*/ $(\tau\iota)_{\tau\omega}$ —an empirical function (attribute) assigning dependently on possible worlds/times a number to an individual; $>/(\alpha\tau\tau)$.

The following construction is a concept of concrete buildings older than 15 years:

$$\lambda w \lambda t [\lambda x [[[]^0\text{Concrete } {}^0\text{Building}]_{wt} x] \wedge [{}^0 > [{}^0\text{Age_of } x] {}^0 15]]].$$

The concepts represented by Examples 1 and 4 are empirical concepts: they construct the intension denoted by the respective NL expression. Logic and concepts cannot do more: the role of empirical concepts is not exhausted by the construction of the intension. We mostly want to find particular objects that satisfy the intension in the actual world-time, and this task can be no more fulfilled by logic, conceptually: it is an empirical task that can be realized only by 'sending a probe into reality'. In our Example 4 the concept constructs the *property* of being a concrete building older than 15 years. Understanding this concept means knowing the *criteria* of seeking the respective buildings. Thus if somebody does not possess the concept of concrete (s)he will not be able to fulfill the task of finding such buildings.

Any ontology registering empirical objects can contain only *intensions* (or rather *concepts of intensions*), but not their values in a particular world/time pair. The latter

¹² For semantic pre-processing of anaphoric references see [7].

are facts contained in a particular database state. The role of ontology is to specify general *conditions* not their satisfiers.

Some concepts contain as their constituents empirical concepts but are not themselves empirical. This is the case of *analytical* concepts, constructing a constant function that returns the same value at every world-time pair. Consider the analytic sentence

“If the building X is higher than the building Y then the building Y is lower than the building X .”

Here we use (simple) empirical concepts ${}^0\text{Higher}$, ${}^0\text{Lower}/(\text{ou})_{\text{to}}$ of the relations-in-intension between individuals, but what is constructed is just the truth-value **T** for any world-time pair. No empirical investigation on the state of the world is needed. The result is unambiguously given due to the meanings/concepts themselves. It is determined only by linguistic convention that the two concepts necessarily satisfy the equality

$$[{}^0\text{Higher}_{wt} x y] = [{}^0\text{Lower}_{wt} y x]$$

for all valuations of w , t , x , y . Thus it might be useful to store in an ontology the necessary rule-like fact that

$$\forall w \forall t \forall xy [{}^0\text{Higher}_{wt} x y] = [{}^0\text{Lower}_{wt} y x].$$

Concepts of many objects (extensions or intensions) are expressed by *simple* (mostly one-word) *expressions*. Here two cases should be distinguished:¹³

(a) The concept expressed by the simple expression is a *simple concept*. Its form is 0X , where X is a non-construction. In this case we are able to identify the constructed object *directly*, i.e., without the mediation of any other concepts.

(b) It is often the case that a simple expression has been introduced into the language as an *abbreviation* of an expression with *complex* meaning, i.e., by a *definition*. This situation is frequent in mathematics. Consider, e.g., the meaning of ‘lattice’, ‘the number π ’, etc. One could hardly usefully work with the real number called ‘ π ’ if (s)he did not know any definition of the number π , like that it is the ratio of the circumference of a circle to its diameter. In an ordinary language it is often the case of idiomatic expressions like ‘bachelor’.

Now it is time to explicate in more details what is meant by a ‘definition’. Above we adduced examples of a definition of the set of prime numbers, of the property of seeking a parking lot, of the real number π , etc. These are *ontological* definitions of the respective *entities*, not of (simple) expressions. On the other hand, when assigning a complex meaning to a simple expression, we define an *expression* by a *linguistic definition*. The next paragraph deals with this difference.

2.4 Ontological vs. Linguistic Definition

Consider the classical equational *linguistic definition* of the form

$$A =_{\text{df}} \Phi(B_1, \dots, B_m),$$

¹³ For details see [16].

where A , called *Definiendum*, is (usually) a simple expression and Φ is a syntactic function creating a complex expression from simple ones; the right-hand side of the definition is called *Definiens*. The role of such a definition consists in associating the *Definiendum* with the meaning that has been associated with the *Definiens*, which can be successfully completed if B_1 through B_m have already been associated with meanings. From the viewpoint of the procedural theory of concepts this means:

Let the meaning of *Definiens* be a complex concept C that is the result of logically analyzing the *Definiens*. Then the concept C becomes a new meaning of the *Definiendum* A .

Moreover, it would not be plausible (though it may happen) that the concept C were a simple concept. Such a definition would not actually introduce any new meaningful expression; it would just name one and the same entity in two different ways. In order to *define* something, the concept C must be an *ontological definition* of the entity E denoted by the *Definiens*, where by ontological definition we mean a *composed* concept of E . If the simple expression E is an abbreviation then there is some definition of a form $E =_{\text{df}} \Phi(B_1, \dots, B_m)$. The concept expressed by $\Phi(B_1, \dots, B_m)$ is thus the respective ontological definition, and this concept is assigned to E as its (new) meaning.

Example 5: The expression *prime numbers*, or *Primes* for short, is an abbreviation.

It has been introduced into mathematics by a definition. Any mathematician, who understands its meaning and wants to utilize it, has to know (at least one of) the respective ontological definition(s) of the set of primes. Let our verbal definitions be as above, i.e.,

$\text{Primes} =_{\text{df}} \text{natural numbers greater than 1 divisible just by itself and by 1.}$

$\text{Primes} =_{\text{df}} \text{natural numbers with just two factors.}$

The respective compound concepts (ontological definitions) of the set of primes are:

$$\lambda x [[^0\text{Nat } x] \wedge [^0 > x \ 1] \wedge [^0 \forall \lambda y [[^0 \text{Div } x \ y] \supset [[x = ^0 1] \vee [x = y]]]]]$$

$$\lambda x [[^0\text{Nat } x] \wedge [^0 \text{Card } \lambda y [^0 \text{Div } x \ y]] = ^0 2]].$$

There are, of course, many other (equivalent) definitions of primes. The question which of the possible definitions was used when the expression ‘primes’ had been introduced into mathematics is not for us to decide, and it is not important from our point of view. What matters, however, is the fact that we must know at least one of them to qualify as competent with respect to ‘primes’. In other words, we can baptise the set of primes ‘primes’, ‘prôtos’, ‘euthymetric’, ‘rectilinear’, or whatever name has been used, but without a complex procedure yielding the set as output, these names are futile. If we connected the expression ‘primes’ only with the simple concept $^0\text{Prime}$ we would end up with the *actual infinity* of the set of primes. But such an infinite set is not accessible to us as a whole, by the one-step Trivialisation. It is accessible only *potentially*, using *analytic ontological definitions* of it.

Similarly we define intensions, in particular properties, using *empirical ontological definitions*. This is often the case of taxonomies. People certainly used perfectly well expressions like ‘dog’ and ‘cat’ without knowing the exact biological taxonomy. They had an intuitive criterion for deciding whether this or that animal is a dog (cat). However, once biologists introduced their classification and defined a dog as an animal of phylum Chordate, class Mammals, order Carnivorous, family Canidae, genus Canis and species Canis Familiaris, it is not an empirical fact that dogs are mammals. We do not have to empirically investigate particular instances of the

property of being a dog in a given state of the world in order to know that necessarily, *ex definitione*, whatever individual happens to be a dog it also instantiates the property of being a mammal. In other words, it is an *analytical fact* that dogs are mammals.

Thus it is, of course, useful to include into our ontology also empirical ontological definitions of particular intentions, or, at least analytical statements expressing necessary relations between properties. They are often called ‘*ISA hierarchies*’.

In this way we explicate the meaning of vague or just intuitively understood expressions by means of previously introduced precise concepts. A new precisely defined concept (*explicatum*) is introduced in place of one which is familiar but insufficiently precise (*the explicandum*). The process of explicating or refining the meanings has to be finished in some bottom level. Otherwise we would end up with an infinite chain of circular definitions. The choice of the simplest concepts that are to be precisely understood and not further refined depends on the *conceptual system* in use.

3 Conceptual Systems

The theory of conceptual systems has been introduced in [17]. Each conceptual system is unambiguously determined by the set of simple concepts that are assigned to simple lexica of a given language by linguistic convention. However, a conceptual system is defined independently of a language as follows:

Let a finite set **Pr** of simple concepts¹⁴ C_1, \dots, C_k be given. Let **Type** be an infinite set of types induced by a finite base (e.g. $\{1, 0, \tau, \omega\}$). Let **Var** be an infinite set of variables, countably infinitely many for each member of **Type**. Finally, let **C** be the set of rules defining *constructions*. Using **Pr**, **Type**, **Var** and **C** an infinite class **Der** is defined as the transitive closure of all the closed complex constructions derivable from **Pr** and **Var** using the rules of **C**, so that:

1. Every member of **Der** is a compound (i.e. non-simple) concept.
2. If $C \in \mathbf{Der}$, then every subconstruction of C that is a simple concept is a member of **Pr**.

Then the set of concepts $\mathbf{Pr} \cup \mathbf{Der}$ is a *conceptual system* based on **Pr**. The members of **Pr** are *primitive concepts*, the members of **Der** are *derived concepts* of the given conceptual system. Evidently, **Pr** unambiguously determines **Der**. The *expressive power* of the given (stage of a) language **L** is then defined as the set **Pr** of the conceptual system underlying **L**. The greater is the expressive power the greater is the area of objects (i.e., the set of objects definable in the conceptual system—constructed by $\mathbf{Pr} \cup \mathbf{Der}$) that can be talked about.

Remark: More precisely, this holds only if **Pr** is ‘independent’, i.e., if **Der** does not contain such a concept C that constructs the same object as a member C' of **Pr**, unless C contains C' as a sub-concept. In other words, the set of primitive concepts of an independent **Pr** is minimal. A simple example of two conceptual systems of propositional logic one of which is and the other is not independent: $\mathbf{Pr}_1 = \{^0\neg, ^0\wedge\}$, $\mathbf{Pr}_2 = \{^0\neg, ^0\wedge, ^0\vee\}$; the latter is not independent, since one of the members of \mathbf{Der}_2 is $\lambda pq [^0\neg [^0\wedge [^0\neg p][^0\neg q]]$, which does not contain $^0\vee$ and constructs the same function as $^0\vee$. Obviously, the area of \mathbf{Pr}_2 is the same (i.e., not greater) as the area of \mathbf{Pr}_1 .

¹⁴ That is Trivializations of non-constructual entities of types of order 1.

Any *natural language* is a ‘living monster’. When we conduct logical analysis we usually examine some *professional language*, i.e., a standardised subclass of a natural language. However, there is not a strict demarcation line between a professional language and a colloquial/natural language. If a student says that spiders are insects, what kind of error has (s)he committed? Is it a linguistic or empirical error, or an analytic (or even logical) error? Is the language of biology (or entomology, etc.) simply a part of the given natural language? So our student does not know English properly? The answer depends on *which* language is used here. It is useful to distinguish what could be called *colloquial language* from *specialized languages*, which indeed share much of their grammar (perhaps all grammar) with the given colloquial language but have a rich lexical part going essentially beyond the lexical part of the colloquial language. Such languages contain, of course, empirical as well as analytic sentences. If the student speaks just a colloquial language, then the meaning of ‘*spider*’ is simply ⁰*Spider*, the Trivialisation of the property of being a spider. And the mistake is just an empirical one. But if the student is learning biology (entomology) and uses a specialised language, then his/her mistake can be compared with the error of some speaker of colloquial English who would claim that spinsters are old widows. He/she does not know the respective *definition* of spiders and the error is an analytical one. The specialised language of entomology uses a richer conceptual system.

4 Necessities

In the outlet of this paper we said that the content of ontology should be a stable part of the system. The question arises how to understand the term ‘stable part’. It should certainly include particular definitions, as explained above, since these definitions give rise to analytically valid statements like “No bachelor is married”, “Cars are vehicles”, “If some x is greater than y then this y is smaller than x ”, “The requisite (necessary condition) of finding an x is the existence of the x ”, etc. These statements express *analytical necessities* (*AN*) that can be defined as follows:

A concept $C/*_n \rightarrow o_{\tau_0}$ is an *analytically necessary* (*AN*) concept iff the proposition identified by C is the proposition *TRUE* that takes the value **T** in all worlds/times. Formally,

$$[{}^0AN {}^0C] = \forall w \forall t [{}^0True_{wt} C],$$

where $AN/(o*_n)$ is the class of closed constructions of order n , $True/(oo_{\tau_0})_{\tau_0}$ is the property of propositions of being true (in a world w at time t).

Thus we have, for instance (*Bachelor*, *Married*, *Car*, *Vehicle*/($o1$) $_{\tau_0}$):

$$\forall w \forall t [{}^0True_{wt} [\lambda w \lambda t \forall x [[{}^0Bachelor_{wt} x] \supset \neg [{}^0Married_{wt} x]]]],$$

$$\forall w \forall t [{}^0True_{wt} [\lambda w \lambda t \forall x [[{}^0Car_{wt} x] \supset [{}^0Vehicle_{wt} x]]]].$$

These analytical necessities have often the form of general *analytical rules* that should be, of course, stored in ontology. They concern particular *intensions* not their extensional values in a given state of the world. We also say that there is a *requisite* relation between particular intensions. Thus if P , Q are constructions of properties of type ($o1$) $_{\tau_0}$ then we have $[{}^0Req Q P]$, meaning that, for all w , t , x , if P is true of x at w , t then so is Q . In database terminology we deal with the *ISA* relation (P implies Q) due to which *inheritance* is established: whatever individual happens to be my car it

inherits all the attributes of a vehicle, because it *is* a vehicle. However, the contingent fact that the individual identified by the registration number 3T45722 is my car is not the fact to be recorded in ontology.

There are, however, *empirical rules* as well. For instance, the laws of nature are not analytical; from the logical point of view, there is no analytical necessity in the fact that, e.g., people cannot levitate (being on the Earth). Yet there is a sort of necessity: providing the laws of nature are valid then there is no time in our world at which a man could levitate. The analysis of this constraint is thus of the form $(x \rightarrow \iota; \text{Man}, \text{Levitate}/(\text{o}\iota)_{\tau\omega}; \text{Situating_on}/(\text{o}\iota\iota)_{\tau\omega}; \text{Earth}/\iota)$

$$\lambda w \forall t [\forall x [[{}^0\text{Man}_{wt} x] \wedge [{}^0\text{Situating_on}_{wt} x {}^0\text{Earth}]] \supset \neg[{}^0\text{Levitate}_{wt} x]].$$

This is the case of the so-called nomological or *nomical necessity*. By ‘nomological or nomic necessity’ we understand the sort of necessity that pertains to laws of nature. Nomic necessity is logically contingent, so the source of the universality that any kind of necessity requires is another one. We obtain universality by suspending temporal variability, which means that it is true (false) at all instants of time that if so-and-so then so-and-so, or that a certain equality between magnitudes obtains. For instance, for all times t , for all individuals x , if x is hot air at t , then x rises at t . This seems to correspond to the sound intuition that the laws of nature are always the same, yet might logically speaking have been different.¹⁵ Nomic necessity is a law-bound correlation between two empirical properties or two propositions (states-of-affairs, events), such that if one is exemplified or obtains then so must the other, or between two magnitudes such that they are equal at all t . Thus, the definition of nomic necessity is as follows:

A concept $C/*_n \rightarrow (\text{o}\iota)_{\tau\omega}$ is a *nomically necessary* (NN) concept iff the proposition that an individual x is C is true in a set of worlds at all times for all the individuals. Formally,

$$[{}^0\text{NN} {}^0C] = \lambda w \forall t [\forall x [{}^0\text{True}_{wt} \lambda w \lambda t [C_{wt} x]]],$$

where $\text{NN}/(\text{o}*_{\tau\omega})$ is the class of closed constructions of order n , $\text{True}/(\text{o}\text{o}_{\tau\omega})_{\tau\omega}$ is the property of propositions of being true (in a world w at time t).

What is constructed is a set of worlds; namely, the set of worlds at which it holds for all times and all individuals that C . This does not exclude the logical possibility of counter-legals, only not within this set of worlds. So the Closure arguably succeeds in pinning down at least one essential feature of nomic necessity. To generalise a bit, we can introduce the type $((\text{o}\text{o}_{\tau\omega})\omega)$ as the type of propositional properties — given a world, we are given a set of propositions; to wit, those eternally true in the given world. One such empirical property is the property of being a nomically necessary proposition. Thus, the Closure $\lambda w [\lambda p [\forall t [p_{wt}]]]$ constructs a function from worlds to sets of propositions. The set is the set of propositions p that are eternal at a given world. Nomically necessary propositions constitute a subset of this set (see [18]).

Some laws are phrased as *generalizations*: “As a matter of nomic necessity, all F ’s are G ’s”. Others are phrased as *equations*: “As a matter of nomic necessity, the magnitude M is proportional to the magnitude N ”. The best-known example of the latter is probably Einstein’s 1905 equation of mass with energy,

$$E = mc^2.$$

¹⁵ However, we bracket the question whether theoretical physics will eventually bear out this assumption.

It bears witness to the thoroughgoing mathematization of physics that the syntactic form of the formula does not reveal that the proportion between energy and mass times the speed of light squared is empirical. Assuming that the special theory of relativity is true, Einstein's discovery of this equivalence was an empirical one. What he discovered was the physical law that two particular magnitudes coincide or are proportional to one another. A unit of energy will be equal to the result of multiplying a unit of mass by the square of the constant of the speed of light. So his equation will be an instance of the following logical form:

$$\lambda w [\forall t [M_{wt} = N_{wt}]].$$

Types: $M, N \rightarrow \tau_{\tau\omega}$ (i.e., magnitudes); $=/(\sigma\tau\tau)$.

When making explicit the empirical character of $E = mc^2$, it is obvious that E , m must be modally and temporally parameterized. But so must c . Though a constant value, the value is constant only relative to a proper subset of the space of all logically possible worlds. It is a logical possibility that in at least some nomologically deviant universe light will have a different velocity.¹⁶ Einstein's equation is constructible thus:

$$\lambda w [\forall t [[^0Mult\ m_{wt} [^0Square\ c_{wt}]] = E_{wt}]].$$

Types: $Mult(lication)/(\tau\tau\tau)$; $Square/(\tau\tau)$; $=/(\sigma\tau\tau)$; $E, m \rightarrow \tau_{\tau\omega}$; $c/\tau_{\tau\omega}$.

What is constructed is the set of worlds at which it is eternally true that E_{wt} is identical to the result of multiplying m_{wt} with the square of c_{wt} . It is the necessity due to some empirical law. Providing the law is valid its consequences are valid as well.

5 The Content of Ontology

Recalling and summarising, the system ontology should encompass:

- a) *Vocabulary*, or *terminology*; this is the specification of the names of *primitive concepts* of the conceptual system in use. As we demonstrated above, we cannot refine particular definitions for ever, we have to choose some basic set of primitive concepts, that are not further refined. This set determines the area of entities (and their types) that the system deals with.

In our model of the traffic system we decided that this part has to contain at least the concepts of *atomic actions*. Atomic action is defined as an agent's activity that can be executed without interaction with the agent's brain. Each atomic action has a unique name and an executive method assigned.

- b) *Ontological definitions* of the most important entities the system deals with. These definitions are compound concepts composed of the primitive concepts specified by the terminological part of ontology. Note that we do not define particular individuals (or instances); rather, *intensions* are defined, most frequently properties.

If needed, *linguistic verbal definitions* can be included here as well, introducing shorthand terms as names of frequently used compound concepts.

¹⁶ Though we acknowledge that essentialists about the velocity of light will claim that c is the same value for all logically possible physical universes. This is not to say that light will travel at the speed of c in all logically possible universes; for at some of them light will not travel at all or light will be missing altogether. So it still constitutes a non-trivial, empirical discovery that the speed of light is c and not any other numerical value.

- c) *Integrity constraints*. Here we specify two kinds of constraints: *analytical* and *empirical*. Note that it is highly recommended to keep these constraints separated. Analytical constraints are valid independently of the state of the world, unlike empirical ones.

Concerning the former, these are *analytically necessary concepts*, mostly specifying *ISA hierarchies* between the properties defined *ad (b)*. As for empirical constraints, we concentrate on *nomologically necessary concepts*. These can encompass also law-like consequences of particular norms and conventions accepted as valid in a given system.

For instance, traffic laws are certainly not valid eternally. However, providing the law is valid its consequences are valid as well. It is useful to specify here constraints the validity of which is supposed to be the case at least during the life-cycle of the system. Thus, for instance, in our traffic model we can have a rule like “*In a crossroads intersecting roads without priority, priority from the right-hand side is applied*”.

- d) *Attributive part*. Here we specify concepts of the most important attributes of the entities specified *ad (a)* and *(b)*. Attributes are empirical functions of a type $(\alpha\beta)_{\tau\omega}$, where α , β are types of (tuples of) entities, see, e.g., [11]. Again, we do not specify particular instances of these functions, which is a matter of a database/knowledge base state. Rather, we define here which important attributes are assigned to the entities of interest; in particular we should specify the identifying attributes.

Besides descriptive attributes like ‘*the registration number of a car*’ we specify here also the relational attributes, in particular the *part-whole relation*. But we must be aware of the fact that the part-whole relation is of a different type than ISA relation; whereas the latter is applied to intensions, the former to individuals. ISA relation is a necessary one and establishes inheritance, unlike the part-whole relation.

For instance, by analytical necessity, all cars are vehicles. However, it is by no way analytically necessary that a particular car consists of an engine, a chassis, and a body. An equally plausible answer to a question on how many parts this particular car consists of might be given in terms of a much longer list: several spark plugs, several pistons, a starter, a carburettor, four tyres, two axles, six windows, etc. It is a common place that a car can be decomposed in several alternative ways. Put it in other words, a car can be constructed in a very simple way as a mereological sum of three things, or in a more elaborate way as a mereological sum of a much larger set of things.

The level of granularity we decide to apply depends on the area investigated. In other words, it depends on the set of primitive concepts we decided to include into our ontology. As stated above, *being a part of* is a property of *individuals*, not of intensions. Thus there is, in general, no inheritance, no implicative relation, between the respective properties ascribed to individual parts and a whole. There is, of course, another question, namely *which parts are essential* for an individual in order to have the *property P*? For instance, an engine is a part of a car, but an engine is not a car. But the property of having an engine is essential for the property of being a car, because something designed without an engine does not qualify as a car but at most as a toy car, which is not a car. But the property of having this or that particular screw, say, is not essential for the property of being a

car.¹⁷ Therefore, if we specify mereological composition of particular entities, we should focus in particular on those parts that are essential for the entity in order to have a property that it happens to have and for which it is of interest in the system.

5.1 Ontology languages and MAS

The last thing we want to mention is a *form* of ontology. In general, any language may serve to encode the procedures, concepts we are interested in. However, as we illustrated in [10], the expressive power of standard web ontology languages is limited by the first-order logic paradigm, which is often not sufficient.

When building up a multi-agent system (MAS), the need for a powerful language of communication becomes even more striking. MAS is a system composed of autonomous, intelligent but resource-bounded agents who are active in their perceiving environment and acting in order to achieve their individual as well as collective goals. As a whole, the system of collaborative agents is able to deal with the situations that are hardly manageable by an individual agent or a monolithic centralised system. The agents communicate and collaborate with each other by exchanging messages formulated in a standardised natural language. According to the FIPA standards¹⁸ for MAS, a *message* is the basic unit of communication. It can be of an arbitrary form but it is supposed to have a structure containing several attributes. Message semantic *content* is one of these attributes, the other being for instance ‘Performatives’, like ‘Query’, ‘Inform’, ‘Request’ or ‘Reply’. The content can be encoded in any suitable language. The FIPA standard languages (for instance the SL language or KIF)¹⁹ are mostly based on the First-Order Logic (FOL) paradigm, enriched with higher-order constructs wherever needed.

The enrichments extending FOL are well-defined syntactically, while their semantics is often rather sketchy, which may lead to communication inconsistencies. Moreover, the bottom-up development from FOL to more complicated cases yields the versions that do not fully meet the needs of MAS communication. In particular, agents’ attitudes and anaphora processing create a problem. Thus using these standards, we cannot easily express all the semantically salient features that should be expressed, like modalities, i.e., particular kinds of necessities or contingency, the distinction between an intensional and extensional level, as we illustrated in this study.

Our agents are ‘born’ with some basic skills; they have built-in methods for basic atomic actions and they also immediately boot up the key-ontology concepts into their internal knowledge base in order to be able to communicate and make decisions. However, they also have to be able to *learn* new concepts, by asking the other “better educated” agents. To this end we need the procedural features of TIL. Agents’ knowledge concerns primarily particular concepts, i.e., constructions (hyper-intensionally individuated procedures), not only their (intensional/extensional) outputs. This is the problem of agents’ attitudes. An agent *a* can easily know or be informed that an obstacle *X* occurs 300 m to the left of him/her without knowing that they themselves is situated 300 m to the right of *X*. Or, *a* may be informed that he/she has to drive around the obstacle *X* without having the *ontological definition* of the property of

¹⁷ This problem is connected with the analysis of property modification, including *being a malfunctioning P*, and we are not going to deal with it here.

¹⁸ The Foundation for Intelligent Physical Agents, <http://www.fipa.org>

¹⁹ For the Semantic content Language SL, see, e.g., <http://www.fipa.org/specs/fipa00008/>; for Knowledge Interface Format KIF, see <http://www-ksl.stanford.edu/knowledge-sharing/kif/>.

‘driving around’ in his/her internal knowledge base. In such a case *a* consults the other agents asking for refining the *concept* of driving around.

Traditional languages like KIF or SL deal with these attitudes in a syntactic way. The agent is related to a piece of syntax. However, it is not a piece of syntax an agent knows but its *semantic content*, i.e., the respective construction encoded by the piece of syntax. Our agents should behave in the same way independently of the language in which their knowledge and ontology is encoded. To this end TIL constructions are needed, since the respective TIL construction is what synonymous expressions (even of different languages) have in common. For instance, if we switch to Czech, the underlying constructions are *identical*:

$${}^0[\lambda w \lambda t [{}^0\text{Driving_around}_{wt} X]] = {}^0[\lambda w \lambda t [{}^0\text{Objetí}_{wt} X]].$$

Thus TIL hyper-intensional features make it possible to build up multi-lingual systems and ontologies without the need to re-program particular methods, providing the *content* of ontologies in particular languages is isomorphic.

6 Conclusion

The above described approach and methods are currently being implemented in the *TIL-Script* programming language, the computational FIPA compliant variant of TIL.²⁰ It is a declarative functional language that serves for encoding the content of ontologies, knowledge bases as well as communication of agents. TIL-Script comprises all the higher-order features of TIL, as the hyperintensional logic of partial functions with procedural semantics and explicit intensionalisation and temporalisation, making thus a communication of software-agents smooth and very natural. TIL constructions are encoded by natural-language expressions in a near-isomorphic manner and for the needs of real-world human agents TIL-Script messages are presented in a standardised natural language. *Vice versa*, humans can formulate their requests, queries, etc., in the standardised natural language that is transformed into TIL-Script messages. Thus the provision of services to humans can be realised in a form close to human understanding.

The development of TIL-Script as well as ontology languages is still a work in progress. Currently we combine traditional tools and languages like OWL (Ontology Web Language), logic programming inference tools (Prolog) and FOL proof calculi (Gentzen system and natural deduction) with the full-fledged features of TIL-Script by building transcription bridges.²¹ Thus the implementation of TIL-Script inference machine proceeds in stages. In the first stage we implemented the subset of language corresponding to the expressive power of Horn clauses. Then we extended it to the full FOL inference machine. The next stage is to implement the inference machine for the subset of classical λ -calculi, and finally, the hyper-intensional features and partiality are to be taken into account. Thus currently TIL-Script serves as a specification language. In future we believe to realise the TIL-Script inference machine in its full expressive power.

²⁰ For details see Ciprich, Duží, Košinár: ‘The TIL-Script language’; in this proceedings.

²¹ For details see Ciprich, Duží, Frydrych, Kohut, Košinár: ‘The Architecture of an Intelligent Agent in MAS’; in this proceedings.

Acknowledgements. This research has been supported by the grant agency of Czech Academy of Sciences, project No. GACR 401/07//0451 “Semantisation of Pragmatics”, and by the program ‘Information Society’ of the Academy of Sciences of CR, project No. 1ET101940420 “Logic and Artificial Intelligence for multi-agent systems”.

References

- [1] Bealer, G. (1982): *Quality and Concept*. Clarendon Press, Oxford.
- [2] Bolzano, B. (1837): *Wissenschaftslehre*. Sulzbach.
- [3] Church, A. (1956): *Introduction to Mathematical Logic I*. Princeton.
- [4] Cocchiarella, N.B. (2003) ‘Conceptual realism and the nexus of predication’. *Metalogicon*, vol. 16, 45-70.
- [5] Cresswell, M.J. (1985): *Structured meanings*. MIT Press, Cambridge, Mass.
- [6] Duží, M. (2004): Concepts, Language and Ontologies (from the logical point of view). In *Information Modelling and Knowledge Bases XV*, Kiyoki,Y., Kangassalo,H., Kawaguchi,E. (eds), IOS Press Amsterdam, 193-209
- [7] Duží, M. (2008): TIL as the Logic of Communication in a Multi-Agent System. *Research in Computing Science*, vol. 33, 27-40.
- [8] Duží, M., Ďuráková, D., Děrgel, P., Gajdoš, P., Müller, J. (2007): Logic and Artificial Intelligence for Multi-Agent Systems. In *Information Modelling and Knowledge Bases XVIII*, Duží, M., Jaakkola, H., Kiyoki, Y., Kangassalo, H. (eds), IOS Press Amsterdam, 236-244.
- [9] Duží, M., Ďuráková, D., Menšík, M. (2005): Concepts are Structured meanings. In *Information Modelling and Knowledge Bases XVI*, Kiyoki,Y., Wangler,B., Jaakkola,H., Kangassalo,H. (eds), IOS Press Amsterdam, 258-276
- [10] Duží, M., Heimburger A. (2006): Web Ontology Languages: Theory and practice, will they ever meet? In *Information Modelling and Knowledge Bases XVII*, Kiyoki Y., Hanno, J., Jaakkola, H., Kangassalo, H. (eds), IOS Press Amsterdam, 20-37
- [11] Duží, M., Materna P. (2003): Intensional Logic as a Medium of Knowledge Representation and Acquisition in the HIT Conceptual Model. In *Information Modelling and Knowledge Bases XIV*, Kangassalo,H., Kawaguchi,E. (eds), IOS Press Amsterdam 51-65.
- [12] Gödel, K. (1990): Russell’s mathematical logic. (1944) In: S. Feferman et alii, eds.: *Kurt Gödel, Collected works, Vol.II.*, Oxford University Press, 120-141.
- [13] Jespersen, B. (2008): Predication and Extensionalization. *Journal of Philosophical Logic*, January 2008. Available at: <http://www.springerlink.com/content/q31731r311428343/?p=60281b936ca249278691b0f25135cc0a&pi=2>
- [14] Kauppi, R. (1967): *Einführung in die Theorie der Begriffssysteme*. Acta Universitatis Tamperensis A/15, Tampere.
- [15] Materna, P. (1998): *Concepts and Objects*. Acta Philosophica Fennica, Vol. 63, Helsinki.
- [16] Materna, P. (1999): ‘Simple concepts and simple expressions’. *Logica ’99*, FILOSOFIA, Prague 2000, 245-257.
- [17] Materna, P. (2004): *Conceptual Systems*. Logos Verlag, Berlin.
- [18] Materna, P. (2005): ‘Ordinary Modalities’. *Logique & Analyse* Vol. 48, No. 189-192. pp. 57-70
- [19] Moschovakis, Y. N. (1994): ‘Sense and denotation as Algorithm and Value’. In J. Väänänen and J. Oikkonen, eds., *Lecture Notes in Logic*, #2 (1994), Springer, 210-249.
- [20] Palomäki, J. (1994): *From Concepts to Concept Theory*. Acta Universitatis Tamperensis A/41, Tampere.
- [21] Tichý, P. (1968): ‘Sense and Procedure’ (“Smysl a procedura”), reprinted in Tichý (2004, pp. 77-92)
- [22] Tichý, P. (1969): ‘Intensions in Terms of Turing Machines’. *Studia Logica* 26, 7-25, reprinted in: Tichý (2004) pp. 93-109.
- [23] Tichý, P. (1988): *The Foundations of Frege’s Logic*, Berlin, New York: De Gruyter.
- [24] Tichý, P. (2004): *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago.

Conceptual Modeling of IS-A Hierarchies for XML

Martin NECASKY and Jaroslav POKORNY
Charles University, Czech Republic

Abstract. In this paper we briefly describe a new conceptual model for XML called XSEM. It is a combination of several approaches in the area. It divides the conceptual modeling process to conceptual and structural level. At the conceptual level, we design an overall conceptual schema of a domain independently on the required hierarchical representations of the data in XML documents. At the structural level, we design required hierarchical representations of the modeled data in different types of XML documents. In this paper, we further extend XSEM for modeling IS-A hierarchies. We also show how conceptual XSEM schemes can be represented at the logical level in XML schemes specified in the XML Schema language.

Keywords. XML, XML Schema, conceptual modeling, IS-A hierarchies

Introduction

XML has recently become an important format for data representation. It is widely applied as a database model, data exchange format, etc. This is mainly because of its simplicity, versatility, and platform independence. As XML data becomes more complex and mission critical the importance of its precise description at the logical and conceptual level grows as well. We can use XML schema languages such as XML Schema to describe XML data. Even though these languages provide powerful constructs for describing structure they are not however suitable for conceptual modeling of XML data. A conceptual model must allow to describe the data independently of any representation. From this point of view, XML schema languages are rather logical level languages than conceptual models. It is similar to the world of relational data where we describe structure of relations at the logical level with relational schemes but model them at the conceptual level with the E-R model or some of its variants. For XML data we therefore need an equivalent of the E-R model for its modeling at the conceptual level.

However, XML data model has some special features such as irregular structure, ordering, mixing structured and unstructured data, or hierarchical structure that are hard to model with the E-R model. Therefore, it is necessary to provide new approaches suitable to model these special features. We provide a survey of this area in [6] where we identify two groups of approaches.

The approaches in the first group, such as [2], [9], or [10], are based on extending E-R with new modeling constructs. However, modeling the required hi-

erarchical structure is problematic with these approaches. It is modeled either by a special type of hierarchical relationship types or by relationship types restricted to binary relationship types with a cardinality $1 : N$. In [8] an algorithm for transforming general E-R schemes (containing $M : N$ and n -ary relationship types) to hierarchical XML schemes is proposed. Similarly, in [1] an algorithm for transforming ORM schemes to hierarchical XML schemes is proposed. The algorithms derive XML schemes from conceptual schemes automatically.

However, there is a common situation where we need to represent concepts such as books and their authors in two or more different hierarchical structures. The first structure can be a list of books and for each book a list of authors and the second structure can be a list of authors and for each author a list of his or her books. It depends on user requirements which hierarchical representations of the concepts are suitable for our system. These hierarchical representations can not be therefore neither determined automatically by the system nor specified by explicit hierarchical relationship types in the conceptual schema by the designer because the modeled semantics of the data would be hidden among a huge number of hierarchical relationship types that have no conceptual meaning.

The approaches in the second group emerge from hierarchical structure. Conceptual schemes are trees whose nodes are entity types and edges are relationship types. The most advanced model in this group was proposed in [4]. The model is called ORA-SS and allows to model generally n -ary relationship types which is not allowed by any other conceptual model in this group to our best knowledge. In the previous example with books we can model each required hierarchy with a separate hierarchical conceptual schema. However, while in E-R or its extensions we can model books by one entity type, in this approach books are represented by two different nodes, one in each schema. Consequently, the modeled semantics is hidden in the hierarchical structures.

The weak points of these approaches result from modeling the semantics and hierarchical structure of the data at the same level. Therefore, it is necessary to further extend recent conceptual models to be suitable for modeling XML data. In [5] we proposed a new conceptual model for XML data called *XSEM*. The basic idea of XSEM, that tries to precede the problems mentioned above, is to divide the modeling process to two levels called *conceptual* and *structural* level. While the semantics of the data is modeled at the conceptual level, the hierarchical representation of the data in XML documents is modeled at the structural level. Therefore, XSEM preserves the advantages of E-R, mainly its simplicity and clearness, and adds the ability to model how the data is represented in different hierarchical XML structures.

Related Work and Contributions. In this paper, we further extend XSEM with constructs for modeling IS-A hierarchies for XML. We also show how to represent XSEM schemes with IS-A hierarchies at the logical XML schema level. Authors of other conceptual models for XML, such as [4], [10] have also studied modeling IS-A hierarchies. However, their interest in modeling IS-A hierarchies was only marginal and insufficient. An exception is the model proposed in [3]. The authors also propose how to model IS-A hierarchies for XML at the conceptual level and how to represent them at the logical XML schema level. For modeling IS-A hierarchies at the conceptual level they extend their own conceptual model

called C-XML. However, C-XML does not divide the conceptual modeling process to the conceptual and structural level as XSEM. It models the semantics as well as structure of the XML documents in one schema. Therefore, we need to augment the approaches such as the one proposed in [3] to be applicable to our approach. It includes modeling IS-A hierarchies at the conceptual level and specification of their representation in different types of XML documents at the structural level. Such approach has not been studied in recent literature yet to our best knowledge.

[3] shows that the XML Schema language does not allow to describe XML structures that can be modeled at the conceptual level using some types of IS-A hierarchies. We discuss these cases later in the paper. To solve this problem, [3] extends XML Schema with new modeling constructs. In our approach, we show that the types of IS-A hierarchies that can not be expressed with XML Schema can be expressed as integrity constraints on XML documents expressed as XQuery expressions. Therefore, we do not need to extend XML Schema.

The paper is organized as follows. In Section 1, we provide a motivating example. Section 2 contains a brief description of XSEM. In Section 3, we extend the conceptual level of XSEM for modeling IS-A hierarchies. In Section 4, we show how to represent IS-A hierarchies at the structural level. In Section 5, we show how to represent conceptual XSEM schemes at the logical XML level in an XML schemes using the XML Schema language complemented with XQuery for some more advanced constraints. We conclude in Section 6.

1. Motivation

Assume a logistic company that ensures transports of goods between destinations. There are several types of transports depending on the mean of transport used. We consider ground and air transports. Figure 1 shows an XML document representing a ground transport and air transport to a target destination. The destination is represented by a root XML element **target** and the transports are represented by the child XML elements **transport** (the ground transport) and **air-transport** (the air transport).

The structure of the XML document can be described with an XML schema language like XML Schema. In this paper, we show how to model such XML document at the conceptual level and how to translate the conceptual description to an XML Schema representation. We are especially interested in modeling IS-A hierarchies. At the conceptual level, an IS-A hierarchy contains different types of concepts where some types are specializations of some other more general types. For example, *transport* is a general type of concepts in our example and there are some types specializing it such as *ground transport* or *air transport*. The specializing types have the same properties as the general type and can have some additional ones.

The XML document at Figure 1 demonstrates how transports organized in the IS-A hierarchy can be represented in XML. Ground transports are represented as XML elements **transport** whereas air transports are represented as XML elements **air-transport**. We assume that there is an XML schema that describes the structure of the XML document. There are the types of contents of **transport**

<pre> <target> <code>AIR-PRG</code> <gps>50N, 14E</gps> <transport> <code>T4211</code> <distance>324</distance> <item> <position>1</position> <code>PKCT271728</code> <size>200x211x122</size> </item> <driver> <licence-no>AM888222</licence-no> <driven-kms>728913</driven-kms> </driver> <truck> <code>5A38882</code> <load-space>60</load-space> </truck> </transport> </pre>	<pre> <air-transport> <code>T4212</code> <distance>7882</distance> <item> <position>1</position> <code>PKCT282213</code> <size>1002x722x1238</size> </item> <captain> <licence-no>82991100022</licence-no> <flight-hours>17911</flight-hours> </captain> <aircraft> <code>TDZ222100</code> <load-space>330</load-space> </aircraft> </air-transport> </target> </pre>
---	---

Figure 1. Demonstration of IS-A hierarchies in XML documents

and `air-transport` XML elements described in this XML schema. Both types have a common part which is composed of XML elements `code`, `distance`, and `item` (repeated once or more times). This common part can be described in a separate general type of content which is extended by the types for the XML elements `transport` and `air-transport` with some additional XML elements (to simplify the explanation, we do not consider XML attributes in the paper). Therefore, IS-A hierarchies at the XML schema level can be comprehended as references of extending types of content of XML elements to previously defined types. In this paper, we are interested in how to specify the representation of conceptual level IS-A hierarchies at the XML schema level. There is not only one possible representation but several. For example, Figure 1 demonstrates just one of the possible XML representations of transports. However, there are other possibilities as well and it depends on user requirements which ones are suitable.

2. XSEM model

XSEM is a new conceptual model for XML. It allows to model special features of XML such as irregular structure, ordering, mixing structured and unstructured data, and hierarchical structure. On the other hand, it is a conceptual model. Therefore, it must abstract from the required XML representations and allow designers to concentrate purely on the semantics of the modeled data.

To achieve these two goals, XSEM is divided to two parts called XSEM-ER and XSEM-H. Similarly to other conceptual models, such as the well-known Entity-Relationship model (E-R), we use XSEM-ER to model real-world objects and relationships between them. At the conceptual level, it is not important how these concepts are represented in XML documents. An XSEM-ER schema describes purely the semantics of the modeled data. On the other hand, we require

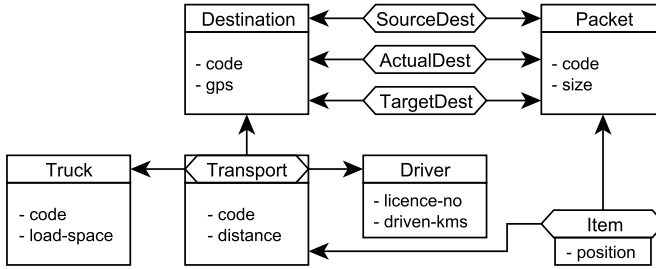


Figure 2. XSEM-ER Schema

the data modeled by the XSEM-ER schema to be represented in one or more types of XML documents. These types of XML documents are modeled at the structural level with XSEM-H.

2.1. XSEM-ER model

XSEM-ER is based on the E-R model. We use entity and relationship types for modeling real-world objects and relationships between them and some additional constructs for modeling the special features of XML data. These extending constructs are not important for the purposes of this paper. For their detailed description we refer to [5]. An example XSEM-ER schema is shown at Figure 2 modeling a part of a domain of a logistic company.

Entity types are used for modeling real-world objects. Each entity type is characterized by its name and zero or more attributes. We further distinguish *strong* and *weak* entity types. *Strong entity types* model objects, whose existence does not depend on other objects. A strong entity type is displayed by a box with the name in the box and attributes under the box. For example, there is a strong entity type *Packet* modeling transported packets.

A *weak entity type* models objects whose existence depends on other objects that are modeled by entity types that are assigned to the weak entity type as so called *determinants*. It is displayed by a box with an inner hexagon. The name is displayed in the box and attributes under the box. It is connected with each of its determinants by a solid arrow oriented to the determinant. For example, there is a weak entity type *Transport*. It models transports of packets. For each transport there must be a target destination, truck used for the transport, and driver who drove the transport. Therefore, *Transport* has three determinants - *Destination*, *Truck*, and *Driver*.

An entity type E models a set of *instances* of E . This set is denoted E^C . For each attribute A of E with a domain $dom(A)$ there is a partial function that assigns to instances of E values of A from $dom(A)$. The value of A assigned to an instance e of E is denoted $A : e$. Further, if E is weak then each instance e of E has assigned an instance of each of the determinants of E . It specifies that the existence of e depends on the existence of the assigned instances of the determinants. The instance of a determinant D of E assigned to e is denoted $D : e$.

Secondly, there are *relationship types*. Relationship types are used for modeling relationships between real-world objects. Each relationship type is charac-

terized by its name, zero or more attributes, and two or more entity types that form the relationship type. These entity types are called *participants* of the relationship type. A relationship type is displayed by a hexagon with the name in the hexagon and attributes under the hexagon. It is connected with each of its participants by a solid arrow oriented to the participant. For example, there is a relationship type *Item* connecting the entity types *Transport* and *Packet*, i.e. the entity types are participants of *Item*. It models relationships between transports and packets, concretely that packets are items of transports. The example shows only relationship types with two participants. However, there can be relationship types with three or more participants as well.

Similarly to entity types, a relationship type R models a set of *instances* of R . This set is denoted R^C . For each attribute A of R with a domain $dom(A)$ there is a partial function that assigns to instances of R values of A from $dom(A)$. Again, $A:r$ denotes the value of A assigned to an instance r of R . Moreover, each instance of R has assigned an instance of each of the participants of R . $P:r$ denotes the instance of a participant P of R assigned to an instance r of R .

2.2. XSEM-H model

An XSEM-ER schema models the semantics of data independently on representation of the data in XML documents. On the other hand, there can be several types of XML documents that represent different parts of the data in different hierarchical structures. To model these types of XML documents we use the XSEM-H model. For each required type of XML documents we design a separate XSEM-H schema. An XSEM-H schema targets one or more entity and relationship types from the XSEM-ER schema and describes how their instances are represented in the modeled XML documents. It does not model any additional semantics of the data. Therefore, XSEM-H schemes are called *views* on XSEM-ER schemes.

XSEM-ER is non-hierarchical because of non-hierarchical weak entity and relationship types. For each of these types there are several possibilities of its hierarchical representation. For example, assume the relationship type *Item* with the participants *Transport* and *Packet*. It is an $M:N$ relationship type. It means that each transport can have more items and each packet can be an item in more transports (packets are not transported directly to target destinations but through more destinations using more different transports). We can require a hierarchical structure where we have a list of transports and for each transport a list of its items, i.e. packets that are items of the transport. We can also require the reversed structure, i.e. to have a list of packets and for each packet a list of transports in which the packet was transported.

In both cases we represented the same relationship type *Item* but in different hierarchical structures. The situation is more complicated if we consider relationship or weak entity types with more than two participants or determinants, respectively, such as *Transport* for example. One of the required representations can be to have a list of transports and for each transport to have the target destination, truck, and driver. Another representation can be to have a list of target destinations and for each target destination to have a list of transports to this destination. For each transport we further want to have the truck and driver. There are also other possibilities of hierarchical representations of *Transport*.

2.2.1. Hierarchical projections

It is not enough to specify required hierarchical representations of weak entity and relationship types in such informal way. For their formal description we propose a formalism called *hierarchical projection*.

Definition 2.1:

A *hierarchical projection* of a relationship or weak entity type T is an expression $T^{T_1, \dots, T_k}[P \rightarrow Q]$ where T_1, \dots, T_k , P , and Q are participants or determinants, respectively, of T . P or Q (exclusively) can be T itself. P is called *parent*, Q is called *child*, and the sequence T_1, \dots, T_k is called *context*.

To describe our previous hierarchical representations of *Item* and *Transport* we use hierarchical projections. We start with *Item*. Hierarchical projections

$$Item[Transport \rightarrow Packet] \quad (H1)$$

$$Item^{Transport}[Packet \rightarrow Item] \quad (H2)$$

describe a hierarchical structure where we have a list of transports and for each transport we have a list of transported packets (*H1*). For each packet in the transport we further need the item relationship connecting the packet with the transport (*H2*). Similarly, we can describe the reversed representation by

$$Item[Packet \rightarrow Transport] \quad (H3)$$

$$Item^{Packet}[Transport \rightarrow Item] \quad (H4)$$

describing a hierarchical structure where we have a list of packets and for each packet we have a list of transports in which the packet was transported (*H3*). For each transport in the packet we further need the item relationship connecting the transport with the packet (*H4*). Another hierarchical representation of *Item* is described by

$$Item[Item \rightarrow Transport] \quad (H5)$$

$$Item[Item \rightarrow Packet] \quad (H6)$$

describing a hierarchical structure where we have a list of item relationships and for each item relationship we have the transport (*H5*) and packet (*H6*) connected by the relationship.

We also discussed hierarchical representations of *Transport*. It is a weak entity type with three determinants. Therefore, specification of its hierarchical representations is more complex. For example, the first representation, that we discussed previously, is described by hierarchical projections

$$Transport[Transport \rightarrow Destination] \quad (H7)$$

$$Transport[Transport \rightarrow Truck] \quad (H8)$$

$$Transport[Transport \rightarrow Driver] \quad (H9)$$

describing a hierarchical structure where we have a list of transports and for each transport we have its target destination (*H7*), truck (*H8*), and driver (*H9*). The second representation is described by

$$\text{Transport}[\text{Destination} \rightarrow \text{Transport}] \quad (H10)$$

$$\text{Transport}^{\text{Destination}}[\text{Transport} \rightarrow \text{Truck}] \quad (H11)$$

$$\text{Transport}^{\text{Destination}}[\text{Transport} \rightarrow \text{Driver}] \quad (H12)$$

describing a hierarchical structure where we have a list of destinations and for each destination we have a list of transports to this destination (H10). For each transport we further have its truck (H11) and driver (H12). Another representation is described by

$$\text{Transport}[\text{Truck} \rightarrow \text{Driver}] \quad (H13)$$

$$\text{Transport}^{\text{Truck}}[\text{Driver} \rightarrow \text{Transport}] \quad (H14)$$

$$\text{Transport}^{\text{Truck, Driver}}[\text{Transport} \rightarrow \text{Destination}] \quad (H15)$$

describing a hierarchical structure where we have a list of trucks and for each truck we have a list of drivers who drove the truck (H13). For each driver we have a list of transports driven by the driver (H14). Because (H14) has the context *Truck* we consider also the superior truck for each driver and we therefore get only a list of transports driven by the driver in this truck. Because the driver can drive transports in more trucks we get a different list of transports for each of these trucks. Finally, for each transport we have the target destination (H15).

2.2.2. XSEM-H views

Hierarchical projections allow to formally describe various hierarchical representations of weak entity and relationship types. However, it is not sufficient to provide designers only with hierarchical projections for modeling hierarchical structure because they require to work in a graphical environment. Therefore, we introduce so called *XSEM-H views*. In this section, we describe only the basic form of XSEM-H views. There are also some additional constructs offered that allow to model more complex hierarchical structures. However, their description is out of the scope of this paper. We refer to [5] for their more detailed description.

Definition 2.2:

Let \mathcal{ER} be an XSEM-ER schema. An *XSEM-H view* \mathcal{H} is a set of oriented trees. Each *node* in \mathcal{H} represents an entity or relationship type from \mathcal{ER} . A node can have assigned a label. Each *edge* in \mathcal{H} represents a hierarchical projection of a weak entity or relationship type from \mathcal{ER} . It is oriented from the *parent node* to the *child node*. \mathcal{H} can not have an arbitrary structure. Let e be an edge in \mathcal{H} with the parent N_p and child N_c . Let e represent a hierarchical projection $T^{T_1, \dots, T_k}[P \rightarrow Q]$. The following conditions must be satisfied:

- (1) N_p represents P and N_c represents Q
- (2) if $k > 0$ then there is an edge having N_p as the child and representing a hierarchical projection $T^{T_1, \dots, T_{k-1}}[T_k \rightarrow P]$; if $k = 0$ then there is no edge having N_p as the child and representing a hierarchical projection of T

These conditions ensure that the structure of the XSEM-H view corresponds to the structure described by the hierarchical projections represented by the edges.

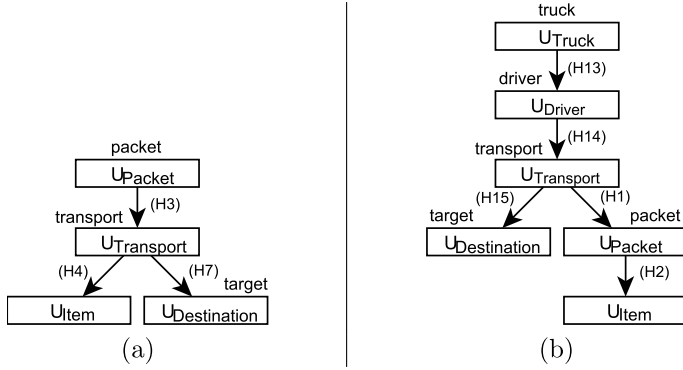


Figure 3. XSEM-H views

A node can have assigned a cardinality constraint in a given edge. The cardinality constraints are formally specified on hierarchical projections. However, we do not describe cardinality constraints on hierarchical projections in a more detail in this paper. For a detailed information we refer to [5].

XSEM-H views are visualized in a similar way we visualize graphs. A node is displayed by a rectangle with the node name in the rectangle. If the node has a label than it is displayed above the node. An edge is displayed by a solid arrow oriented from the parent node to the child node.

Figure 3 shows two example XSEM-H views on the XSEM-ER schema at Figure 2. They model two different types of XML documents representing transports. Each node in the views is denoted by U_{Type} where $Type$ is the entity or relationship type represented by the node. Each edge is labeled with the hierarchical projection it represents. We use only the numbers of the hierarchical projections. They were specified in Section 2.2.1.

An XSEM-H view \mathcal{H} specifies how instances of entity and relationship types represented by the nodes in \mathcal{H} are represented in XML documents. The hierarchical structure is formally given by the hierarchical projections represented by the edges of \mathcal{H} . \mathcal{H} itself only adds some supplemental information that is not provided by hierarchical projections but is necessary for the specification of the modeled structure of XML documents. This includes names of XML elements that are given by the labels assigned to the nodes in \mathcal{H} and ordering on XML elements that is given by the ordering on the edges going from the corresponding node.

Formally, let U_T be a node in \mathcal{H} representing T . Let t be an instance of T . U_T specifies that t is represented as a sequence of XML elements. The value of an attribute A of T assigned to t is represented as an XML element with the name and type given by A . The XML elements representing the values of the attributes are ordered in the order given by T and followed by representations of the instances of the entity and relationship types assigned to t by the edges having U_T as the parent. The ordering on the representations of the instances follows the ordering described by U_T . If U_T has assigned a label, the XML representation of t is enclosed into an XML element with the label as its name.

The following XML document has the structure described by the view (a). It is an XML representation of an instance of the entity type *Packet* as described by the view. The root node U_{Packet} has a label *packet* and models the XML element **packet**. The attributes of the entity type *Packet* model the XML elements **code** and **size**. Further, there is the edge going from U_{Packet} to $U_{Transport}$. $U_{Transport}$ has a label *transport*. It models the XML elements **transport** in **packet**. The attributes of *Transport* model the corresponding XML elements in **transport**. There are also XML elements **position** that are modeled by the attribute *position* of *Item*. Because U_{Item} has not assigned a label the XML code modeled by U_{Item} is not encapsulated in a separate XML element. Instead, it is included in the XML code modeled by the parent node $U_{Transport}$. Therefore, **position** are child XML elements of **transport**.

<packet>	<code>DST281</code>	<target>
<code>PCKT83821</code>	<gps>50N,14E</gps>	<code>DST237</code>
<size>100x23x211</size>	</target>	<gps>55N,12E</gps>
<transport>	</transport>	</target>
<code>T3821</code>	<transport>	</transport>
<distance>872</distance>	<code>T4783</code>	</packet>
<position>18</position>	<distance>245</distance>	
<target>	<position>27</position>	

3. IS-A hierarchies in XSEM-ER

In the E-R model extended with IS-A hierarchies we can create an entity type *S* that models some additional semantics to the semantics modeled by another entity type *G*. The additional semantics is modeled with attributes of *S* and relationship types having *S* as a participant. At the instance level, each instance *s* of *S* is also an instance of *G*. It means that *s* has assigned not only values of the attributes of *S* but also of *G*. We use this mechanism in XSEM-ER. For modeling IS-A hierarchies we use so called *IS-A relationship types*.

Definition 3.1:

Let *G* and *S* be two entity types. An *IS-A relationship type* is a pair $(G, S)_{IS-A}$ where *G* and *S* are called *general* and *specializing* entity type, respectively. We can also say that *G* *generalizes* *S* or that *S* *specializes* *G*. At the instance level, the IS-A relationship type specifies that $S^C \subseteq G^C$

Because each instance *s* of *S* is also an instance of *G* it has assigned values of attributes of *G* and also determinants of *G*, if *G* has any. Therefore, *S* is not described only by its own attributes and determinants but also by the attributes and determinants of *G*. We call these attributes and determinants *inherited* attributes and determinants from *G*.

There can be another entity type *G'* and an IS-A relationship type $(G', S)_{IS-A}$. In such case we say that there are *multiple generalizations* for *S*. It means that *S* inherits attributes and determinants from *G* and *G'* as well.

We can also require several constraints to be held by general and specializing entity types. Assume entity types *G*, *S*₁, ..., *S*_{*n*} and IS-A relationship types

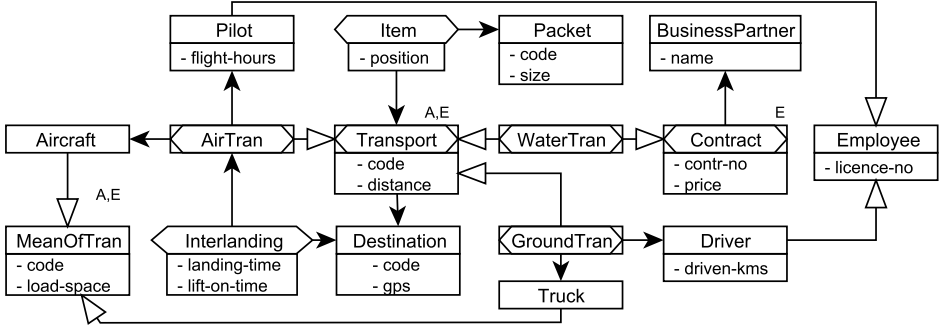


Figure 4. XSEM-ER: IS-A hierarchies

$(G, S_1)_{IS-A}, \dots, (G, S_n)_{IS-A}$. Without any constraints $\bigcup_{i=1}^n S_i^C \subseteq G^C$. Further, there can be $1 \leq i, j \leq n$ such that $S_i^C \cap S_j^C \neq \emptyset$.

Firstly, we can require $\bigcup_{i=1}^n S_i^C = G^C$ to be held. It means that each instance of G must be also an instance of some of S_1, \dots, S_n . This constraint is usually called *union constraint*. We specify it by denoting G as *abstract*. Secondly, we can require $S_i^C \cap S_j^C = \emptyset$ for some i, j where $i \neq j$, $1 \leq i, j \leq n$. We allow only a simpler constraint where we require S_1^C, \dots, S_n^C to be mutually exclusive, i.e. $\forall i, j$ where $i \neq j$ and $1 \leq i, j \leq n$ we require $S_i^C \cap S_j^C = \emptyset$. This constraint is usually called *mutual-exclusion constraint*. We specify it by denoting G as *exclusive*.

We display IS-A relationship type by an empty arrow oriented from the specializing entity type to the general entity type. If the general entity type is denoted as abstract or exclusive then we mark the box with A or E , respectively. Figure 4 shows an example of IS-A hierarchy. There is the weak entity type *Transport* with a determinant *Destination*. The entity type models transports in general. We further distinguish air, ground, and water transports modeled by *AirTran*, *GroundTran*, and *WaterTran*, respectively, that specialize *Transport*. *Transport* is an abstract and exclusive entity type. It means that a union constraint for *Transport* and mutual-exclusion constraint for *AirTran*, *GroundTran*, and *WaterTran* must hold. There are also other IS-A hierarchies in the schema. For example, the entity type *Contract* models contracts with external business partners and water transports are contracts with external business partners. Therefore, *WaterTran* specializes *Contract*. This is an example of multiple generalizations because *WaterTran* specializes *Transport* as well.

4. IS-A hierarchies in XSEM-H

In this section, we show how IS-A relationship types are represented in XSEM-H views. Suppose firstly that we construct an XSEM-H view that represents an entity type E and there is one or more entity types generalizing E . To describe the hierarchical representation of E we can specify hierarchical projections for the determinants of E as well as its inherited determinants.

Assume for example the XSEM-H view at Figure 5. It represents the entity type *WaterTran* that inherits determinants from *Transport* and *Contract*. Its

hierarchical representation was constructed on the base of hierarchical projections $WaterTran[WaterTran \rightarrow Destination]$ and $WaterTran[WaterTran \rightarrow BusinessPartner]$.

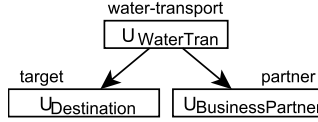


Figure 5. XSEM-H views

Secondly, suppose that we construct an XSEM-H view that represents an entity type G . We therefore construct an XSEM-H view \mathcal{H}_G from G on the base of hierarchical projections of G . The resulting \mathcal{H}_G describes how instances of G are represented in XML documents. Further suppose an entity type S that specializes G . Because $S^C \subseteq G^C$, \mathcal{H}_G also describes the XML representation of instances of S . However, S extends G with some additional determinants whose hierarchical representation is not described with \mathcal{H}_G . Therefore, we can require to construct an additional XSEM-H view \mathcal{H}_S that extends \mathcal{H}_G . To denote that \mathcal{H}_S extends \mathcal{H}_G we use so called *IS-A* edges.

Definition 4.1:

Let U_G and U_S be two nodes in an XSEM-H view \mathcal{H} . Let U_S be a root in \mathcal{H} . Let the nodes represent entity types G and S , respectively. Let $(G, S)_{IS-A}$. An *IS-A edge* is a pair $(U_G, U_S)_{IS-A}$ where U_G is called *general* node and U_S is called *specializing* node of the IS-A edge. We also say that U_G *generalizes* U_S and U_S *specializes* U_G .

The IS-A edge specifies that instances of G are represented in XML documents according to the structure of the node U_G . If an instance of G is also an instance of S then it is represented in XML documents according to the structure of the node U_G complemented with the structure of the node U_S . Moreover, if U_S has a label than this label is used instead of the label of U_G .

The condition (2) from Definition 2.2 is not applied to the edges having U_S as the parent before the completion. It must be satisfied after the completion. When we reformulate the condition, the following must hold. Let e be an edge with the parent U_S and representing a hierarchical projection $S^{T_1, \dots, T_k}[S \rightarrow Q]$. If $k > 0$ then there must be an edge having U_G as the child and representing a hierarchical projection $G^{T_1, \dots, T_{k-1}}[T_k \rightarrow G]$ (because S is generalized to G we must replace S with G). If $k = 0$ then there can not be an edge having U_G as the child and representing a hierarchical projection of G .

In our graphical representation, an IS-A edge is displayed by an empty arrow oriented from the specializing node to the general node. Assume the example XSEM-H view at Figure 6. It shows how to model the XML representation of the IS-A edges from the XSEM-ER schema at Figure 4. This representation was demonstrated by the XML document at Figure 1. It contains an XSEM-H view $\mathcal{H}_{Transport}$ representing the entity type $Transport$. It is composed of nodes

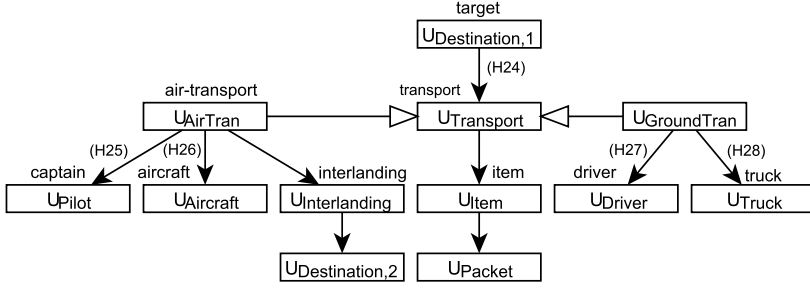


Figure 6. XSEM-H: IS-A hierarchies

$U_{Destination,1}$ and $U_{Transport}$. These nodes are connected by the edge that represents a hierarchical projection $Transport[Destination \rightarrow Transport]$ (H24).

Further, there are XSEM-H views $\mathcal{H}_{AirTran}$ and $\mathcal{H}_{GroundTran}$ representing the entity types *AirTran* and *GroundTran*. $\mathcal{H}_{AirTran}$ is composed of nodes $U_{AirTran}$, U_{Pilot} , and $U_{Aircraft}$ and two edges representing hierarchical projections $AirTran^{Destination}[AirTran \rightarrow Pilot]$ (H25) and $AirTran^{Destination}[AirTran \rightarrow Aircraft]$ (H26). Similarly, $\mathcal{H}_{GroundTran}$ is composed of nodes $U_{GroundTran}$, U_{Driver} , and U_{Truck} and two edges representing hierarchical projections $GroundTran^{Destination}[GroundTran \rightarrow Driver]$ (H27) and $GroundTran^{Destination}[GroundTran \rightarrow Truck]$ (H28).

The views $\mathcal{H}_{AirTran}$ and $\mathcal{H}_{GroundTran}$ extend the view $\mathcal{H}_{Transport}$ which is denoted by the IS-A edges $(U_{Transport}, U_{AirTran})_{IS-A}$ and $(U_{Transport}, U_{GroundTran})_{IS-A}$. It means that instances of *Transport* are represented in XML documents according to the structure of $U_{Transport}$. However, if an instance of *Transport* is also an instance of *AirTran* or *GroundTran* then it is represented according to $U_{Transport}$ complemented with $U_{AirTran}$ or $U_{GroundTran}$, respectively. Representations of *Transport* and *GroundTran* instances are enclosed into XML elements **transport** ($U_{GroundTran}$ does not have its own label) and representations of *AirTran* instances are enclosed into XML elements **air-transport**. Because the edges representing the hierarchical projections (H25), (H26), (H27), and (H28) complement $\mathcal{H}_{Transport}$ they must have *Destination* as their context. Otherwise, the condition (2) given by Definition 2.2 would not be satisfied after the completion of $U_{Transport}$ with these edges.

5. Derivation of XML schemes

An XSEM-H view describes a given type of XML documents at the conceptual level. We need to derive an XML schema that describes this type at the logical level. To describe types of XML documents at logical level we use XML schema languages. In this section we show how to derive from an XSEM-H view a corresponding logical XML schema in the XML Schema language. For the derivation, we do not consider XML attributes but only XML elements. Firstly, we show a derivation of an XML schema from an XSEM-H view without IS-A edges. Secondly we consider IS-A edges.

5.1. XSEM-H views without IS-A edges

Assume a node U in the XSEM-H view that represents an entity or relationship type T . From U we derive a complex content `<xs:sequence><!-- components --></xs:sequence>`. The complex content has assigned a unique name denoted $type_U$ (not in the XML schema directly, we will use the name later). The components of the complex content are derived from the attributes of T and edges having U as the parent:

1. for each attribute A of T with a domain $dom(A)$ we derive an element declaration `<xs:element name="A" type="dom(A)" />`
2. for each edge E with the parent U and child V ((min, max) is the cardinality of V in E)
 - if V has a label l then we derive an element declaration `<xs:element name="l" type="type_V" minOccurs="min" maxOccurs="max" />`
 - if V has not a label then we derive a group declaration `<xs:group ref="type_V" minOccurs="min" maxOccurs="max" />`

The components of the complex content definition are ordered in the order prescribed by U . Firstly, there are components derived from the attributes in the order prescribed by T . Secondly, there are components derived from the edges in the order prescribed by U .

The complex content derived from U is included in the resulting XML schema in a form of complex type definition or group. If U has a label then it describes XML elements named by this label. The content of these XML elements is described by the complex content derived from U . The complex content must be assigned to the XML elements as a complex type. Therefore, we put into the XML schema a complex type with the derived complex content and with $type_U$ as its name. If U has not a label then it can not describe XML elements because we do not have a name for them. It describes only a group of XML elements derived from its attributes and edges. Therefore, we put into the XML schema a group with the derived complex content and with $type_U$ as id.

Finally, global element declarations for root nodes with labels are inserted into the resulting XML schema. If U is a root node with a label l then we derive a global element declaration `<xs:element name="l" type="type_U" />`.

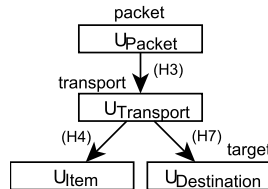


Figure 7. XSEM-H view

We demonstrate the derivation on example. Assume the XSEM-H view at Figure 7. For example, for $U_{Transport}$ we derive a complex type


```

<xs:complexType name="Transport"/>
  <xs:sequence>
    <xs:element name="code" type="xs:string" />
    <!-- element declarations for the other attributes -->
    <xs:group ref="Item" />
    <xs:element name="target" type="Destination" />
  </xs:sequence>
</xs:complexType>

```

We derived a complex type because $U_{Transport}$ has the label *transport*. The element declarations `code` and the commented ones are derived from the attributes of *Transport*. The edge going to U_{Item} is represented as a reference to a group because U_{Item} has no label. The edge going to $U_{Destination}$ is represented as an element declaration because $U_{Destination}$ has a label. For U_{Item} we derive a group:

```

<xs:group id="Item"/>
  <xs:sequence>
    <xs:element name="position" type="xs:string" />
  </xs:sequence>
</xs:group>

```

For U_{Packet} and $U_{Destination}$ we derive corresponding complex types in a similar way. Finally, we derive a global element declaration `<xs:element name="packet" type="Packet" />` for the root node U_{Packet} because it has a label.

5.2. XSEM-H views with IS-A edges

In this section we show how to represent IS-A edges in derived XML schemes. As we showed in Section 5.1 there is a distinction between the representations of nodes with and without labels in XML schemes. To simplify the description of the representation of IS-A edges in XML schemes we however assume that each node has a label. It means that it is represented in the derived XML schema as a complex type. This label can be a label assigned directly to the node or a label inherited from another node. If a node has no label and is specialized by another nodes then it can be represented using **group** content model. However, it is technically complicated even though the idea is similar. Therefore, we think it is better to omit these details for the reader.

To describe the representation of IS-A edges in XML schemes we must distinguish several cases. Without a loss of generality assume nodes U_G , U_S , and $U_{S'}$ representing entity types G , S , and S' , respectively, and IS-A edges $(U_G, U_S)_{IS-A}$ and $(U_G, U_{S'})_{IS-A}$.

Suppose firstly that G is **exclusive**, i.e. $S^C \cap S'^C = \emptyset$. It means that each instance of G that is not an instance of S nor S' is represented according to U_G and each instance of S or S' is represented according to U_S or $U_{S'}$, respectively. Therefore, we derive complex contents from U_G , U_S , and $U_{S'}$ by the basic algorithm in Section 5.1. Moreover, the complex contents derived from U_S and $U_{S'}$ extend the complex content derived from U_G . For this we use **extension** construction in the XML Schema language. It allows to extend an existing complex type (i.e. the one derived from U_G) to new complex types (i.e. the ones derived from U_S and $U_{S'}$).

Because U_G has a label l_{U_G} instances of G are represented as XML elements named l_{U_G} with the structure described by $type_{U_G}$. If U_S has not its own label then it inherits l_{U_G} . It means that instances of S are represented as XML elements named l_{U_G} as well. However, their structure is described by $type_{U_S}$. Therefore, we need to declare XML elements named l_{U_G} whose structure is described by $type_{U_G}$ or $type_{U_S}$. This is described in the derived XML schema already. We have the element declaration `<xs:element name=" l_{U_G} " type=" $type_{U_G}$ " />` derived from U_G . The rules of the XML Schema language specify that each XML element described by this declaration has its content described by $type_{U_G}$ or any of the extended types, i.e. $type_{U_S}$ as well. If U_S has its own label l_{U_S} different from l_{U_G} then instances of S are represented as XML elements named l_{U_S} and not l_{U_G} . It means that XML elements l_{U_S} can appear wherever XML elements l_{U_G} can appear. Therefore, we need to declare XML elements named l_{U_G} with the structure described by $type_{U_G}$ and XML elements named l_{U_S} with the structure described by $type_{U_S}$. Further, we must specify that XML elements l_{U_S} can appear wherever XML elements l_{U_G} can appear. The same must be done for $U_{S'}$.

If U_G is a root node then XML elements l_{U_G} are declared globally and we therefore declare globally XML elements l_{U_S} and $l_{U_{S'}}$, as well. The result are global element declarations for U_G , U_S , and $U_{S'}$. If U_G is not a root node then XML elements l_{U_G} are declared locally in the complex content derived from the parent of U_G . We must specify that there can appear not only XML elements l_{U_G} but also l_{U_S} and $l_{U_{S'}}$. Therefore, we replace the original declaration of XML elements l_{U_G} with

```
<xs:choice minOccurs="min" maxOccurs="max">
  <xs:element name=" $l_{U_G}$ " type=" $type_{U_G}$ " />
  <xs:element name=" $l_{U_S}$ " type=" $type_{U_S}$ " />
  <xs:element name=" $l_{U_{S'}}$ " type=" $type_{U_{S'}}$ " />
</xs:choice>
```

where (min, max) is the cardinality of U_G in the edge going to U_G .

Assume further that G is not only exclusive but also abstract. It means that each instance of G is also an instance of one of its specializations. If S and S' are all of its specializations then each instance of G is represented according to U_S or $U_{S'}$ and none is represented according to U_G . To denote this explicitly in the XML schema we can mark the complex type derived from U_G as abstract using an XML Schema attribute **abstract** set to **true**. We also does not include the element declaration for U_G in the XML schema. On the other hand, if there is moreover a specialization S'' of G that is not represented in the XSEM-H view then instances of S'' are represented according to U_G . Therefore, we can not set the complex type derived from G as abstract and the element declaration for U_G must be included in the XML schema.

Secondly, let us discuss the situation where G is **not exclusive**. It means that there can be an instance of G that is an instance of S and S' at once. We suppose that U_G has a label l_{U_G} . We start with G being not abstract. If U_S and $U_{S'}$ does not have their own labels they inherit the label from U_G . It means that each instance g of G is represented as an XML element l_{U_G} according U_G . If g is an instance of S , S' , or both (G is not exclusive) then the structure of the XML element is complemented with U_S , $U_{S'}$, or both, respectively. It means that we

need a complex type for XML elements l_{U_G} that is composed of the complex content derived from U_G and optionally complemented with the complex contents derived from U_S and $U_{S'}$, i.e.

```
<xs:complexType name="type $_{U_G}$ ">
  <xs:sequence>
    <!-- components for  $U_G$  -->
    <xs:sequence minOccurs="0">
      <!-- components for  $U_S$  -->
    </xs:sequence>
    <xs:sequence minOccurs="0">
      <!-- components for  $U_{S'}$  -->
    </xs:sequence>
  </xs:sequence>
</xs:complexType>
```

If G is moreover abstract and there are no other specializing entity types of G the situation is more complicated. It means that each instance of G must be also an instance of S or S' , or both (G is not exclusive). Therefore, each instance of G is represented as an XML element l_{U_G} according to U_G complemented with U_S , $U_{S'}$, or both. However, the previously generated complex type allows an instance of G to be represented as an XML element l_{U_G} without these completions. We must therefore ensure that one or more of the sequences derived from the specializing nodes is repeated. In our case we get a choice of two variants: the first subsequence has `minOccurs="1"` and the second has `minOccurs="0"` or the first subsequence has `minOccurs="0"` and the second has `minOccurs="1"`. This leads to a very complicated complex type if we consider tens or more specializing entity types. However, there is not a better solution using standardized XML Schema constructs. A better solution is to use a language that allows to describe such more advanced constraints in XML data. Such a language is for example XQuery.

We create the complex type $type_{U_G}$ as in the previous case. For each sequence derived from a specialization of U_G we identify the first element declaration with `minOccurs="n"` where $n \geq 1$, if there is any. Therefore, we get a set of declarations of XML elements. Let e_1, \dots, e_k be names of these XML elements. We check if some of the XML elements is present in the content of each XML element l_{U_G} . The required constraint is then expressed by the following expression ($xpath_{U_G}$ denotes an XPath expression targeting XML elements described by U_G):

```
for      $e in  $xpath_{U_G}$ 
return  if count($e/ $e_1$ )=0 AND ... AND count($e/ $e_k$ ) = 0
        then <validation-error/>
        else <validation-ok />
```

Meanwhile, we supposed that U_S and $U_{S'}$ inherit the label l_{U_G} from U_G . Suppose now that they have their own labels $l_{U_S} \neq l_{U_{S'}}$, i.e. if g is an instance of S than it is represented as an XML element l_{U_S} according to U_S . If it is an instance of S' than it is represented as an XML element $l_{U_{S'}}$ according to $U_{S'}$. The problem is that we require g to be represented as two different XML elements if it is an instance of both S and S' (it is possible because G is not exclusive). Moreover, these XML elements have a common part described by U_G . One possibility is to

represent the common part as an XML element l_{U_G} and the specializing parts as separate XML elements l_{U_S} and $l_{U_{S'}}$. For this we need a mechanism of keys for entity types to interconnect these XML elements (they represent the same instance which must be explicitly denoted in the XML representation). However, modeling XML keys at the conceptual level is a complicated problem and we have no space to describe it in this paper. For a detailed description of modeling XML keys at the conceptual level with XSEM we refer to [7].

To solve this problem without a mechanism of keys we can represent each instance of U_G as an XML element l_{U_G} . If it is also an instance of S or S' we can include directly to the XML element l_{U_G} an XML element l_{U_S} or $l_{U_{S'}}$, respectively, containing the corresponding specializing part of the content. The resulting complex type describing XML elements l_{U_G} is following:

```
<xs:complexType name="type $U_G$ ">
  <xs:sequence>
    <!-- components for  $U_G$  -->
    <xs:element name="l $U_S$ " minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <!-- components for  $U_S$  -->
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- and similarly for  $S'$  -->
  </xs:sequence>
</xs:complexType>
```

If G is abstract we need to specify a similar constraint as in the previous case. Again we can express this constraint with a similar XQuery expression.

We demonstrate the basic ideas of the representation of IS-A hierarchies in XML schemes on examples. Suppose the XSEM-H view in Figure 6. We derive a complex type **Transport** from $U_{Transport}$ by the basic algorithm. Then we derive complex types **AirTran** and **GroundTran** from $U_{AirTran}$ and $U_{GroundTran}$, respectively. Their complex contents are derived by the basic algorithm but the resulting complex types are extensions of **Transport**. **AirTran** is derived as follows (**GroundTran** is derived similarly):

```
<xs:complexType name="AirTran">
  <xs:complexContent>
    <xs:extension base="Transport">
      <xs:sequence>
        <!-- components derived from  $U_{AirTran}$  -->
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Finally, we derive declarations of XML elements **transport** and **air-transport**. Because $U_{Transport}$ has the parent $U_{Destination,1}$, **transport** XML elements must be declared locally in the complex content derived from $U_{Destination,1}$. Further, wherever a **transport** XML element can appear an **air-transport** XML ele-

ment can appear as well. The complex type derived from $U_{Destination,1}$ therefore contains a choice of the declarations of these XML elements:

```
<xs:complexType name="Destination1">
  <xs:sequence>
    <!-- element declarations for the attributes -->
    <xs:choice>
      <xs:element name="transport" type="Transport"/>
      <xs:element name="air-transport" type="AirTran"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Even though the entity type *Transport* is abstract the complex type **Transport** is not denoted as abstract because there is also the entity type *WaterTran* that is not represented in the XSEM-H view and its instances are therefore represented according to $U_{Transport}$, i.e. as XML elements **transport** with the complex type **Transport**.

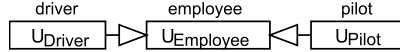


Figure 8. XSEM-H views

Finally, assume the XSEM-H view at Figure 8. It is rather a fragment of a larger XSEM-H view. We use it to demonstrate the representation of general entity types that are not exclusive. This is the case of the entity type *Employee* specialized by *Driver* and *Pilot*. We derive

```
<xs:element name="employee" type="Employee">
<xs:complexType name="Employee">
  <xs:sequence>
    <xs:element name="number" type="xs:string"/>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="driver" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="drivenMiles" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!-- and similarly for U_Pilot -->
  </xs:sequence>
</xs:complexType>
```

If *Employee* was abstract we would need to specify that each **employee** XML element must contain **driver**, **pilot**, or both. This can be expressed by the following XQuery expression complementing the XML schema:

```
for    $e in employee
return if count($e/driver)=0 AND count($e/pilot) = 0
      then <validation-error/>
      else <validation-ok />
```

6. Conclusions

We described a conceptual model for XML called XSEM and we further extended it for modeling IS-A hierarchies. The proposed approach allows to abstract from complex constructions in XML schemes when representing IS-A hierarchies. It allows to start with modeling the semantics of the data at the conceptual level independently on its structural representation. After modeling the semantics, the designer can move to the structural level and specify how the modeled concepts including IS-A hierarchies are represented in several types of XML documents.

We showed how to represent XSEM schemes with IS-A hierarchies at the logical XML schema level using the XML Schema language. We showed that complicated cases of multiple generalizations and specializations unconstrained by union and mutual-exclusion constraints can be represented in XML schemes with standard constructions of XML Schema and XQuery.

In our future work we are going to extend the algorithm translating XSEM schemes to XML schemes. We will enable user to choose from different translation strategies that utilize various XML schema design strategies such as Russian Doll or Salami Slice design.

References

- [1] BIRD, L., GOODCHILD, A., AND HALPIN, T. A. Object Role Modelling and XML-Schema, Conceptual Modeling. In *Proceedings of the 19th International Conference on Conceptual Modeling*. 2000. Springer, Salt Lake City, Utah, USA, 309–322.
- [2] AL-KAMHA, R., EMBLEY, D. W., AND LIDDLE, S. W. Augmenting traditional conceptual models to accommodate xml structural constructs. In *Proceedings of 26th International Conference on Conceptual Modeling*. 2007. Springer, Auckland, New Zealand, 518–533.
- [3] AL-KAMHA, R., EMBLEY, D. W., AND LIDDLE, S. W. Representing Generalization/Specialization in XML Schema. In *Enterprise Modelling and Information Systems Architectures, Proceedings of the Workshop in Klagenfurt*. 2005, LNI 75 GI 2005, Klagenfurt, Germany, 250–263.
- [4] DOBBIE, G., XIAOYING, W., LING, T., AND LEE, M. ORA-SS: An object-relationship-attribute model for semi-structured data. Tech. Rep. December 2000, Department of Computer Science, National University of Singapore, Singapore.
- [5] NECASKY, M. Conceptual Modeling for XML. PhD Thesis, September 2008. Charles University, Prague, Czech Republic. <http://kocour.ms.mff.cuni.cz/~necasky/dw/thesis.pdf>
- [6] NECASKY, M. Conceptual modeling for XML: A survey. Tech. Rep. 3/2006, Charles University, Prague, Czech Republic. 54 p.
- [7] NECASKY, M. AND POKORNY, J. 2007. Extending E-R for Modelling XML Keys.. In *Proceedings of The Second International Conference on Digital Information Management*. IEEE Computer Society. Lyon, France, 236–241.
- [8] PIGOZZO, P. Quintarelli E. An algorithm for generating XML Schemas from ER Schemas. In *Proceedings of the Thirteenth Italian Symposium on Advanced Database Systems*. 2005, Brixen-Bressanone (near Bozen-Bolzano), Italy, 192–199.
- [9] PSAILA, G. ERX: A conceptual model for XML documents. In *Proceedings of the 2000 ACM Symposium on Applied Computing*. 2000, ACM, Como, Italy, 898–903.
- [10] SENGUPTA, A., MOHAN, S., AND DOSHI, R. XER - extensible entity relationship modeling. In *Proceedings of the XML 2003 Conference*. 2003, Philadelphia, USA, 140–154.

Boolean Constraints for XML Modeling

Sven HARTMANN^a, Sebastian LINK^{b,1} and Thu TRINH^c

^a *Department of Informatics, Clausthal University of Technology, Germany*

^b *School of Information Management, Victoria University, New Zealand*

^c *School of Engineering and Advanced Technology, Massey University, New Zealand*

Abstract. The study of integrity constraints has been identified as one of the major challenges in XML database research. The main difficulty is finding a balance between the expressiveness and the existence of automated reasoning tools for different classes of constraints.

In this paper we define Boolean constraints for XML by exploring homomorphisms between XML data trees and XML schema graphs. These constraints are naturally exhibited by XML data due to its nested structure.

We demonstrate, in contrast to many other proposals, that reasoning about Boolean constraints is well-founded. That is, we establish that the interaction between Boolean constraints corresponds precisely to the logical implication of Boolean propositional formulae. Therefore, our Boolean constraints do not only capture valuable semantic information about XML data but also permit reasoning support by off-the-shelf SAT solvers. Finally, we identify a few subclasses of Boolean constraints for which the implication problem can be solved efficiently.

Keywords. XML, Modeling, Semantics, Constraints, Propositional Logic

1. Introduction

The eXtensible markup language (XML,[5]) has evolved to be the standard for data exchange on the Web. Moreover, it represents a uniform model for data integration. While it provides a high degree of syntactic flexibility it has little to offer to specify the semantics of its data. Consequently, the study of integrity constraints has been recognised as one of the most important yet challenging areas of XML research [11]. The importance of XML constraints is due to a wide range of applications ranging from schema design, query optimisation, efficient storing and updating, data exchange and integration, to data cleaning [11]. Therefore, several classes of integrity constraints have been defined for XML including keys, path constraints, inclusion constraints [11] and functional dependencies [2,13,15,18,24,25,26,27,30].

For relational databases around 100 different classes of data dependencies have been studied [23]. However, for each of these classes there is a well-accepted single concept for the type of dependency, e.g. in the database community there is a common understanding of the notion of a functional dependency in relational databases. However, the complex nature of XML data has resulted in various different proposals for functional

¹Corresponding Author: Sebastian Link, School of Information Management, Victoria University, Wellington, New Zealand; E-mail: sebastian.link@vuw.ac.nz.

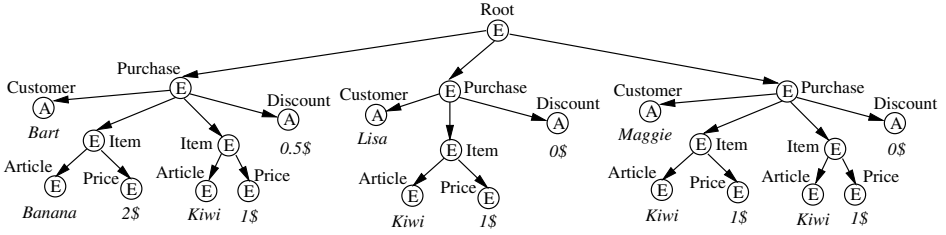


Figure 1. XML data tree exhibiting some functional dependency

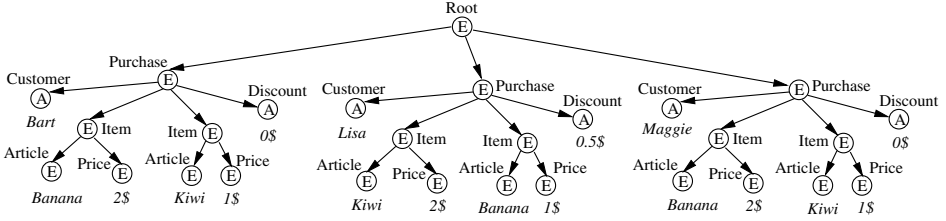


Figure 2. Another XML data tree exhibiting another type of functional dependency

dependencies in the context of XML (XFDs). Many of these XFD classes deviate in their expressiveness but they are all justified since they naturally occur in XML data. To see an example of these different classes we consider the XML data tree in Figure 1. It contains simple purchase profiles that shows customers, the items they bought (an item is a pair consisting of an article and its price) and the discount received for their items bought. In this data tree same articles have the same price. This observation is likely to be called a functional dependency between the article and its price.

Now consider the data tree in Figure 2. It is easy to observe that the functional dependency between an article and its price is no longer valid. Nevertheless, the data stored in this tree is still not independent from one another: whenever two customers have purchased all the same items then they both receive the same discount. That is, the set of items purchased functionally determines the discount. Note that this is not a functional dependency that occurs just accidentally but captures important semantic information that should be satisfied by every legal XML data of this form. Notice that *Lisa* received a discount of 0.5\$ since *Kiwis* for the price of 2\$ were on special.

The majority of proposals has studied the first kind of XFDs [2,18,25] while the second kind has been studied in [13,14,26]. Recently there has been considerable effort into finding a unifying class of XFDs which can capture the majority of previously defined classes [26,27], in particular both types of XFDs described above. While it is desirable to express as much information as possible it is also important to have automated reasoning techniques for these XFDs. This will prove essential for the implementation of native XML database management systems and many applications. Hence, it is important to identify the exact expressiveness of this constraints.

Contributions. In this paper we extend the expressiveness of XFDs that are of the second kind as illustrated in Figure 2. Previous results have shown that the implication of these XFDs is decidable in time linear in the input size of the problem [14]. However, one may argue that many desirable properties cannot be expressed with these constraints. For instance, if we consider an XML view for a sales analysis in which only *different*

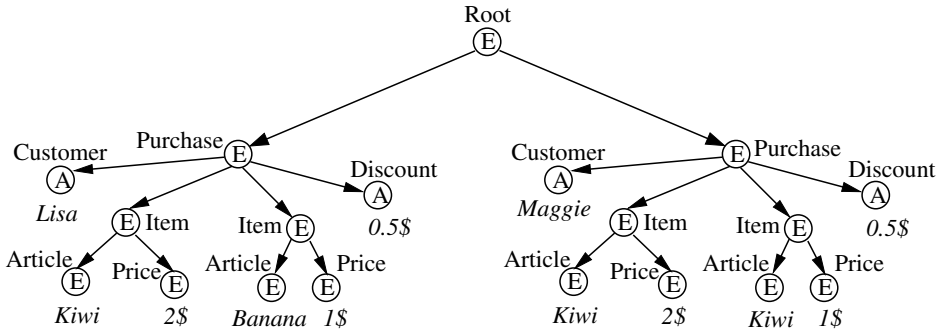


Figure 3. A counterexample for the implication of Rule (iii) by Rules (i) and (ii)

purchases by the same customer are of interest, then the following constraint is satisfied: the same customers in this view cannot have purchases in which all the items are the same (i.e. the purchases must deviate in some item). This dependency cannot be captured by means of a functional dependency.

The first contribution of this article is an extension of the formal definition of an XFD to a broader class of XML constraints which we call Boolean constraints. XFDs of the second type appear as special cases of Boolean constraints. Many desirable properties of XML data that cannot be expressed by other constraints can indeed be modelled by Boolean constraints, e.g. the constraint on the XML view above.

As a second contribution we demonstrate that reasoning about Boolean constraints is well-founded. For instance, imagine a store that implements the following business rules: (i) the same customer receives the same discount (e.g. a loyalty discount) and (ii) customers that purchase all the same items receive the same discount. The store manager is interested whether the following business rule is a consequence of the first two: (iii) whenever the same discount is applied to two purchases, then the purchases belong to the same customer or the purchases consist of exactly the same items. If Rule (iii) follows indeed from Rules (i) and (ii), then Rule (iii) does not need to be specified additionally. However, Figure 3 shows some XML data that satisfies the first two Rules, but violates Rule (iii).

Hence, if Rule (iii) is desired, then it must be specified in addition to Rules (i) and (ii). We show that this type of reasoning can be fully automatised using off-the-shelf state-of-the-art SAT solvers. In fact, we argue that the implication of Boolean constraints corresponds exactly to the logical implication of formulae in propositional logic. This provides a characterisation of the expressiveness of Boolean constraints. As an immediate consequence it follows that the associated implication problem for Boolean constraints is coNP-complete. However, this verifies that they are expressive and that results from artificial intelligence research can be directly applied towards implementing reasoning facilities. Moreover, we identify subclasses of Boolean constraints whose associated implication problem can be decided efficiently.

In summary, Boolean constraints cannot only be utilised to express many properties that naturally appear in XML data but also permit automated reasoning support that can be implemented effectively by native XML database management systems. This is in contrast to many other classes of XML constraints [11].

Related Work. There are many proposals that deal with the first kind of XFDs [2,18,25]. These proposals are all based on a path-like notion of functional dependencies that is reminiscent of earlier research in semantic and object-oriented data models [22,28]. XFDs as introduced by Arenas/Libkin [2] do not enjoy a finite ground axiomatisation and their implication problem is coNP complete for restricted classes of DTDs [2]. However, a restricted class of these XFDs is indeed finitely axiomatisable [15]. Vincent et al. [25] introduced strong functional dependencies for XML, i.e. XFDs of the first kind, provided an axiomatisation for unary dependencies of this class (only a single path is allowed on the left) and showed the associated implication problem to be decidable in linear time [24]. The most expressive class of XFDs has been introduced by Wang/Topor [26,27]. It covers, in particular, the XFDs of the second kind. However, none of the associated decision problems have been investigated for this class.

Finally, Boolean dependencies have been studied for relational databases [20,21]. Our work can be seen as an extension of that work to XML data, and also as an extension of our research on XFDs of the second kind [14].

Organisation. The article is structured as follows. In Section 2 we introduce the basic terminology used throughout the paper. In particular, we review an XML graph model that allows us to model XML data, schemata and constraints. In Section 3 we reveal which schema information is necessary to uniquely identify XML data. This will justify our definition of Boolean constraints which are introduced and illustrated in Section 4. The expressiveness of Boolean constraints is analysed in Section 5. We demonstrate that counterexample trees for the implication of these constraints correspond exactly to truth assignments in propositional logic that serve as a counterexample for an instance of the implication problem for propositional formulae. We identify some subclasses of Boolean constraints in Section 6 whose associated decision problems are decidable efficiently. Finally, we conclude in Section 7 and briefly comment on future work.

2. XML Graphs and Trees

In this paper we follow a fairly simple XML graph model [13] that is still powerful enough to capture different kinds of XFDs. We assume basic familiarity with notions from graph theory such as graphs, trees and paths [16]. Note that all graphs in this paper are considered to be finite, directed and without parallel arcs.

2.1. Rooted Graphs and Trees

A *rooted graph* is a directed acyclic graph $G = (V_G, A_G)$ with a distinguished vertex r_G , called the *root* of G , such that every vertex of G can be reached from r_G by passing a directed path of forward arcs. In a rooted graph every vertex but r_G has at least one predecessor, while r_G has no predecessor. For every vertex v , let $\text{Succ}_G(v)$ denote its (possibly empty) set of successors in G . Vertices without successors are said to be *leaves*. Let $L_G \subseteq V_G$ denote the set of leaves of G . A *rooted tree* is a rooted graph $T = (V_T, A_T)$ where every vertex $v \in V_T$ but the root r_T has exactly one predecessor.

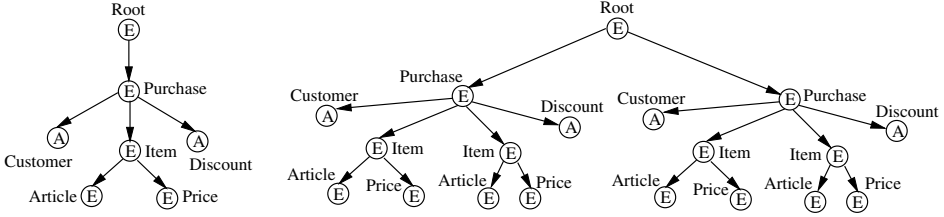


Figure 4. Two XML trees with vertex labels illustrating names and kinds of vertices

2.2. Graphs and Trees for XML

Rooted graphs are frequently used to illustrate the structure of XML documents [2, 11, 18]. In general, we assume that there are fixed sets $ENames$ of element names and $ANames$ of attribute names, and a fixed symbol S indicating text.

An *XML graph* is a rooted graph G together with two mappings $name : V_G \rightarrow ENames \cup ANames$ and $kind : V_G \rightarrow \{E, A\}$ assigning every vertex its name and kind, respectively. If G is, in particular, a rooted tree we also speak of an *XML tree*. The symbols E and A tell us whether a vertex represents an element or attribute, respectively. We suppose that vertices of kind A are always leaves, while vertices of kind E can be either leaves or non-leaves. In addition, we suppose that vertices of kind E have a name from $ENames$, while vertices of kind A have a name from $ANames$.

An *XML data tree* is an XML tree T together with an evaluation $val : L_T \rightarrow STRING$ assigning every leaf a (possibly empty) string, see Figures 1 and 2.

An *XML schema graph* is an XML graph G together with a mapping $freq : A_G \rightarrow \{?, 1, *, +\}$ assigning every arc its frequency. Every arc terminating in vertices of kind A has frequency $?$ or 1 , while arcs terminating in vertices of kind E may have any frequency. We say that an arc a is of *multiple frequency*, if $freq(a) = *$ or $freq(a) = +$. In Figure 5, for example, we marked all arcs with their frequency, except those of frequency 1 . Further, we suppose that no vertex in an XML schema graph G has two successors sharing both their kind and their name. Hence, the first graph in Figure 4 may serve as an XML schema graph, while the second one does not. If G is an XML tree we also speak of an *XML schema tree*.

2.3. Homomorphisms

Let G' and G be two XML graphs, and consider a mapping $\phi : V_{G'} \rightarrow V_G$. We call the mapping ϕ *root-preserving* if the root of G' is mapped to the root of G , that is, $\phi(r_{G'}) = r_G$. Further, ϕ is *kind-preserving* if the image of a vertex is of the same kind as the vertex itself, that is $kind(v') = kind(\phi(v'))$ for all $v' \in V_{G'}$. Finally, ϕ is *name-preserving* if the image of a vertex carries the same name as the vertex itself, that is, $name(v') = name(\phi(v'))$ for all $v' \in V_{G'}$. The mapping ϕ is a *homomorphism* between G' and G if the following conditions hold:

1. every arc of G' is mapped to an arc of G , that is, $v', w' \in A_{G'}$ implies $(\phi(v'), \phi(w')) \in A_G$,
2. ϕ is root-, kind- and name-preserving.

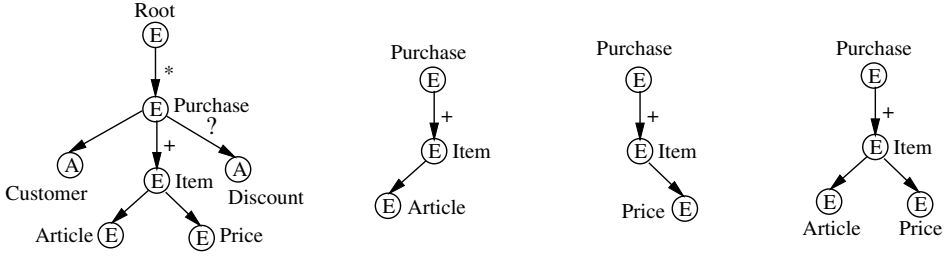


Figure 5. XML schema graph and three v_{purchase} -subgraphs $\llbracket \text{Article} \rrbracket$, $\llbracket \text{Price} \rrbracket$ and $\llbracket \text{Article} \rrbracket \sqcup \llbracket \text{Price} \rrbracket$

A homomorphism ϕ may be naturally extended to the arc set of G' : given an arc $a' = (v', w')$ of G' , $\phi(a')$ denotes the arc $(\phi(v'), \phi(w'))$ of G .

As an example, consider the two XML trees in Figure 4. There is a unique name-preserving mapping ϕ which maps the vertices of the second graph G' to the vertices of the first graph G . (That is, all vertices with name *Purchase* in the second graph are mapped to the single vertex with name *Purchase* in the first graph, etc.) It is not difficult to verify that this mapping satisfies conditions (i) and (ii) above, i.e., ϕ is indeed a homomorphism.

Let T' be an XML data tree and G an XML schema graph. T' is said to be *compatible* with G if there is a homomorphism $\phi : V_{T'} \rightarrow V_G$ between T' and G such that for each vertex v' of T' and each arc $a = (\phi(v'), w)$ of G , the number of arcs $a' = (v', w'_i)$ mapped to a is at most 1 if $\text{freq}(a) = ?$, exactly 1 if $\text{freq}(a) = 1$, at least 1 if $\text{freq}(a) = +$, and arbitrarily many if $\text{freq}(a) = *$. Due to the definition of XML schema graph, this homomorphism is unique if it exists.

For example, consider the XML data tree T' from Figure 1 and the XML schema tree T in Figure 5. Again, the unique name-preserving mapping from $V_{T'}$ to V_T is a homomorphism between T' and T . On comparing both trees and checking the frequencies, it turns out that T' is compatible with T .

A homomorphism $\phi : V_{G'} \rightarrow V_G$ between XML graphs G' and G is an *isomorphism* if ϕ is bijective and ϕ^{-1} is a homomorphism, too. Then G' is said to be *isomorphic* to G or a *copy* of G . We call two isomorphic XML data trees T' and T *equivalent* if the isomorphism $\phi : V_{T'} \rightarrow V_T$ between them is also *evaluation-preserving*, i.e., $\text{val}(v') = \text{val}(\phi(v'))$ holds for every vertex $v' \in V_{T'}$.

2.4. v -subgraphs

Let G be a rooted graph, v a vertex of G and $L \subseteq L_G$ be a set of leaves of G . Consider the union U of all directed paths of G from v to some leaf in L . We call U a v -subgraph of G . Clearly, every non-empty v -subgraph of an XML graph is itself an XML graph. In particular, a non-empty v -subgraph of an XML tree is again an XML tree. If v is clear from the context, then we denote the v -subgraph U as $\llbracket l_1, \dots, l_k \rrbracket$ where $\{l_1, \dots, l_k\} = L_U$. In particular, $\llbracket \emptyset \rrbracket$ denotes the empty v -subgraph. We use $\text{Sub}_G(v)$ to denote the set of all v -subgraphs of G . For example, in Figure 5 we have the three v_{purchase} -subgraphs $\llbracket \text{Article} \rrbracket$, $\llbracket \text{Price} \rrbracket$ and their v -subgraph union $\llbracket \text{Article} \rrbracket \sqcup \llbracket \text{Price} \rrbracket = \llbracket \text{Article, Price} \rrbracket$. We use \sqcap to denote v -subgraph intersection, for example $\llbracket \text{Customer, Article} \rrbracket \sqcap \llbracket \text{Article, Price} \rrbracket = \llbracket \text{Article} \rrbracket$.

If U contains all leaves of G which may be reached from v by passing a directed path of G , then U is said to be the *total v -subgraph* of G and is denoted by $G(v)$. Note that $G(v)$ is maximal among $Sub_G(v)$ and contains every other v -subgraph of G as a v -subgraph itself.

Let G' and G be XML graphs. A homomorphism $\phi : V_{G'} \rightarrow V_G$ between them induces a mapping of the total subgraphs of G' to the total subgraphs of G : given a total v' -subgraph $G'(v')$ of G' , $\phi(G'(v'))$ denotes the total $\phi(v')$ -subgraph $G(\phi(v'))$ of G . Note that the vertices and arcs of $G'(v')$ are mapped to the vertices and arcs of $G(\phi(v'))$. Obviously, the number of pre-images of a total v -subgraph of G coincides with the number of pre-images of the vertex v .

Let G' and G be two XML graphs together with a homomorphism ϕ between them. An $r_{G'}$ -subgraph U' of G' is a *subcopy* of G if U' is isomorphic to some r_G -subgraph U of G , and an *almost-copy* of G if it is maximal with this property. Almost-copies are of special interest as an XML data tree compatible with some XML schema tree T does not necessarily contain copies of T .

2.5. Projections

Let G' and G be XML graphs together with a homomorphism $\phi : V_{G'} \rightarrow V_G$ between them. Given an r_G -subgraph U of G , the *projection* of G' to the subgraph U is the union of all subcopies of U in G' , and denoted by $G'|_U$. In particular, $G'|_U$ is an $r_{G'}$ -subgraph of G' .

3. Identifying XML data

Let T be some XML schema tree, v a vertex of T and T' be some XML data tree compatible with T . In this section we will answer the question which v -subgraphs of T suffice to identify pre-images of $T(v)$ in T' up to equivalence. More precisely, what is the minimal set $\mathcal{B}(v) \subseteq Sub_T(v)$ such that equivalence between two arbitrary pre-images W_1, W_2 of $T(v)$ in T' can be determined by the equivalences of $W_1|_X$ and $W_2|_X$ on all $X \in \mathcal{B}(v)$? In other words, for which minimal $\mathcal{B}(v)$ is it true that if W_1 and W_2 are not equivalent, then there is some $X \in \mathcal{B}(v)$ such that $W_1|_X$ and $W_2|_X$ are already not equivalent?

The set $\mathcal{B}(v)$ should not consist of all v -subgraphs of T since projections on some v -subgraphs uniquely determine projections on other v -subgraphs. For instance, consider the XML schema tree in Figure 5. The projections of any $v_{Purchase}$ -subgraphs on $[[Customer]]$ and on $[[Article]]$ suffice to decide the equivalence of the projections on the union $[[Customer, Article]]$ of $[[Customer]]$ and $[[Article]]$. The reason for this is that the $v_{Purchase}$ -subgraphs $[[Customer]]$ and $[[Article]]$ do not share any arc of multiple frequency. Consequently, the $v_{Purchase}$ -subgraph $[[Customer, Article]]$ cannot belong to $\mathcal{B}(v)$.

In order to answer the initial question we seek some intuition from relational databases. Which subschemata of a relation schema allow us to uniquely identify tuples in a relational database? Suppose we have a relation schema $R = \{A_1, \dots, A_k\}$ with attributes A_1, \dots, A_k . Then every R -tuple $t : R \rightarrow \bigcup_{i=1}^k dom(A_i)$ with $t(A_i) \in dom(A_i)$ is uniquely determined by its projections $\{t(A_1), \dots, t(A_k)\}$. That is, there cannot be

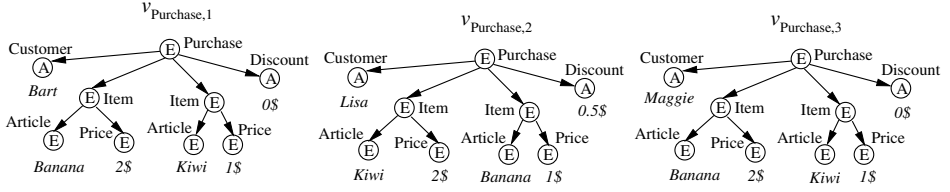


Figure 6. The three pre-images of the total $v_{Purchase}$ -subgraph of the XML schema tree from Figure 5 in the XML data tree T' from Figure 2. All three pre-images have the same projection on $\llbracket Article \rrbracket$, and on $\llbracket Price \rrbracket$, but only the first and third pre-image have the same projection on $\llbracket Article, Price \rrbracket$.

two different tuples which have the same projection on all the attributes. Consequently, the unary subschemata (those consisting of a single attribute) allow us to identify tuples.

With this in mind it is natural to conjecture that $\mathcal{B}(v) = \mathcal{U}(v)$, i.e., $\mathcal{B}(v)$ consists of exactly all unary v -subgraphs of T , i.e., all the directed paths from v to a leaf of T . However, it turns out that this set is too small. We will demonstrate this by the following example.

In fact, Figure 6 shows the three pre-images of the total $v_{Purchase}$ -subgraph of the XML schema tree from Figure 5 in the XML data tree T' from Figure 2. Two unary $v_{Purchase}$ -subgraphs are $\llbracket Article \rrbracket$ and $\llbracket Price \rrbracket$. All three pre-images have the same projection on $\llbracket Article \rrbracket$, and on $\llbracket Price \rrbracket$. However, the second and third pre-image still deviate in their projection on $\llbracket Article, Price \rrbracket$. In this sense, the projection on $\llbracket Article \rrbracket$ and the projection on $\llbracket Price \rrbracket$ do not allow us to distinguish between the second and the third $v_{Purchase}$ pre-image. The reason for this is that $X = \llbracket Article \rrbracket$ and $Y = \llbracket Price \rrbracket$ do share an arc of multiple frequency.

Definition 1 Let T be some XML schema tree, v a vertex of T and $X, Y \in Sub_T(v)$. The v -subgraphs X and Y are called *reconcilable* if and only if there are v -subgraphs X' of X and Y' of Y such that X' and Y' share no arc of multiple frequency and such that $X' \sqcup Y' = X \sqcup Y$ holds. \square

This means, that whenever X and Y share some arc (u, w) of frequency $*$ or $+$, then X contains the total w -subtree of Y or Y contains the total w -subtree of X .

Example 1 The $v_{Purchase}$ -subgraphs $X = \llbracket Article \rrbracket$ and $Y = \llbracket Price \rrbracket$ both contain the arc $(v_{Purchase}, v_{Item})$ while their v_{Item} -subgraphs are obviously no subgraphs of one another; see Figure 5. Hence, $X = \llbracket Article \rrbracket$ and $Y = \llbracket Price \rrbracket$ are not reconcilable. On the other hand, $X = \llbracket Customer \rrbracket$ and $Y = \llbracket Article \rrbracket$ are indeed reconcilable. \square

The next theorem justifies the syntactic definition of reconcilability in semantical terms. In fact, two v -subgraphs X and Y are reconcilable precisely if the projections on X and on Y uniquely determine the projection on $X \sqcup Y$.

Theorem 1 Let T be some XML schema tree, and v a vertex of T . For all $X, Y \in Sub_T(v)$ we have that X and Y are reconcilable if and only if for all XML data trees T' that are compatible with T and for all pre-images W, W' of $T(v)$ in T' the following holds: if $W|_X$ is equivalent to $W'|_X$ and $W|_Y$ is equivalent to $W'|_Y$, then $W|_{X \sqcup Y}$ is equivalent to $W'|_{X \sqcup Y}$. \square

Example 2 Consider the pre-images W rooted at $v_{\text{Purchase},1}$ and W' rooted at $v_{\text{Purchase},2}$ in Figure 6. Their projections on $X = \llbracket \text{Article} \rrbracket$ and on $Y = \llbracket \text{Price} \rrbracket$ are equivalent, but their projections on $X \sqcup Y = \llbracket \text{Article}, \text{Price} \rrbracket$ are not equivalent. This is an instance of Theorem 1 since X and Y are not reconcilable. \square

It now remains to define the set of v -subgraphs that are necessary and sufficient to decide equivalence between arbitrary pre-images.

Definition 2 Let T be some XML schema tree, and v a vertex of T . The set $\mathcal{B}(v) \subseteq \text{Sub}_T(v)$ of essential subgraphs is defined as the smallest set of v -subgraphs of $T(v)$ with the following two properties: (i) $\mathcal{U}(v) \subseteq \mathcal{B}(v)$, and (ii) if $X, Y \in \mathcal{B}(v)$ are not reconcilable, then $X \sqcup Y \in \mathcal{B}(v)$. \square

The essential subgraphs of v form therefore the smallest set that contains the unary subgraphs of v and that is closed under the union of v -subgraphs that are not reconcilable. This seems now very natural: for any $X, Y \in \mathcal{B}(v)$ for which the two projections $W|_X$ and $W|_Y$ of some pre-image W of v do not uniquely determine the projection of $W|_{X \sqcup Y}$, the subgraphs X and Y cannot be reconcilable, and $X \sqcup Y$ is therefore included in $\mathcal{B}(v)$.

Example 3 Let T denote the XML schema tree in Figure 5. Consequently, $\mathcal{B}(v_{\text{Purchase}})$ consists of the following v_{Purchase} -subgraphs:

$\llbracket \text{Customer} \rrbracket$, $\llbracket \text{Article} \rrbracket$, $\llbracket \text{Price} \rrbracket$, $\llbracket \text{Discount} \rrbracket$, and $\llbracket \text{Article}, \text{Price} \rrbracket$.

The binary v_{Purchase} -subgraph $\llbracket \text{Article}, \text{Price} \rrbracket$ belongs to $\mathcal{B}(v_{\text{Purchase}})$ since $\llbracket \text{Article} \rrbracket$ and $\llbracket \text{Price} \rrbracket$ are not reconcilable. It follows that, in order to determine any pre-image of $T(v_{\text{Purchase}})$ up to equivalence, only up to five different projections are necessary. \square

The analysis in this section will give us invaluable information for defining Boolean constraints in the next section.

4. Boolean Constraints for XML

In this section we will use the previous definition of essential subgraphs to introduce Boolean constraints into the context of XML. These extend Boolean dependencies that have been studied for the relational model of data [20,21] and also a certain type of functional dependencies for XML [13,14].

4.1. The formal Definition of Boolean Constraints

For the sake of simplicity we will restrict ourselves to Boolean constraints defined with respect to a fixed XML schema tree T . It is, however, well-known that every rooted graph G may be uniquely transformed into a rooted tree T by unfolding it, i.e., by splitting vertices with more than one predecessor.

The following definition defines the syntactic structure of Boolean constraint expressions.

Definition 3 Let T be an XML schema tree, and $v \in V_T$ a vertex of T . The set of Boolean constraints over the vertex v is defined as the smallest set $BC(v)$ with the following properties:

- if $X \in \mathcal{B}(v)$, then $v : X \in BC(v)$,
- if $v : \varphi \in BC(v)$, then $v : \neg\varphi \in BC(v)$, and
- if $v : \varphi, v : \psi \in BC(v)$, then $v : (\varphi \wedge \psi) \in BC(v)$. □

For the sake of readability we introduce the following abbreviations: $v : (\varphi \vee \psi)$ stands for $v : \neg(\neg\varphi \wedge \neg\psi)$ and $v : (\varphi \Rightarrow \psi)$ stands for $v : (\neg\varphi \vee \psi)$.

Example 4 Consider the XML schema tree in Figure 5. Some Boolean constraints over $v_{Purchase}$ are (we omit outmost parentheses)

- $\varphi_1 = v_{Purchase} : \llbracket Customer \rrbracket$,
- $\varphi_2 = v_{Purchase} : \llbracket Customer \rrbracket \vee (\llbracket Article \rrbracket \wedge \llbracket Price \rrbracket)$,
- $\varphi_3 = v_{Purchase} : \llbracket Customer \rrbracket \vee \llbracket Article, Price \rrbracket$,
- $\varphi_4 = v_{Purchase} : \llbracket Article, Price \rrbracket \Rightarrow \llbracket Discount \rrbracket$, and
- $\varphi_5 = v_{Purchase} : \llbracket Discount \rrbracket \Rightarrow (\llbracket Customer \rrbracket \vee \llbracket Article, Price \rrbracket)$.

The expression $v_{Purchase} : \llbracket Customer, Article, Price \rrbracket \Rightarrow \llbracket Discount \rrbracket$ is not a Boolean constraint over $v_{Purchase}$ since $\llbracket Customer, Article, Price \rrbracket$ is not an essential sub-graph of $v_{Purchase}$. □

The next definition will assign the semantics to Boolean constraints. We will define when a compatible XML tree satisfies a Boolean constraints.

Definition 4 Let T be an XML schema tree, $v \in V_T$ a vertex of T , and T' an XML data tree that is compatible with T . Two distinct pre-images W_1, W_2 of $T(v)$ in T' are said to satisfy the Boolean constraint φ over v , denoted by $\models_{\{W_1, W_2\}} \varphi$, if and only if the following holds:

- if $\varphi = v : X$ for some $X \in \mathcal{B}(v)$, then $\models_{\{W_1, W_2\}} \varphi$ if and only if $W_1|_X$ and $W_2|_X$ are equivalent,
- if $\varphi = v : \neg\psi$ for some $\psi \in BC(v)$, then $\models_{\{W_1, W_2\}} \varphi$ if and only if not $\models_{\{W_1, W_2\}} v : \psi$,
- if $\varphi = v : (\psi_1 \wedge \psi_2)$ for $\psi_1, \psi_2 \in BC(v)$, then $\models_{\{W_1, W_2\}} \varphi$ if and only if $\models_{\{W_1, W_2\}} v : \psi_1$ and $\models_{\{W_1, W_2\}} v : \psi_2$.

We say that T' satisfies the Boolean constraint φ over v , denoted by $\models_{T'} \varphi$, if and only if for all distinct pre-images W_1, W_2 of $T(v)$ in T' we have that $\models_{\{W_1, W_2\}} \varphi$. □

4.2. Some Examples

We will now illustrate Definition 4 on the Boolean constraints $\varphi_1, \dots, \varphi_5$ from Example 4. Let T_1 denote the XML data tree in Figure 1, and T_2 denote the XML data tree in Figure 2. Both T_1, T_2 are compatible with the XML schema tree in Figure 5.

The constraint $\varphi_1 = v_{Purchase} : \llbracket Customer \rrbracket$ expresses the fact that the projections of any two distinct pre-images of $T(v_{Purchase})$ on $\llbracket Customer \rrbracket$ must be equivalent. In other words, an XML data tree satisfies φ_1 when only purchases of the same customer are stored. Both T_1 and T_2 do not satisfy φ_1 .

The constraint $\varphi_2 = v_{\text{Purchase}} : \llbracket \text{Customer} \rrbracket \vee (\llbracket \text{Article} \rrbracket \wedge \llbracket \text{Price} \rrbracket)$ expresses the fact that for any two distinct pre-images of $T(v_{\text{Purchase}})$ the projections on $\llbracket \text{Customer} \rrbracket$ must be equivalent or both projections on $\llbracket \text{Article} \rrbracket$ and on $\llbracket \text{Price} \rrbracket$ must be equivalent. In other words, an XML data tree satisfies φ_2 when only purchases of the same customer are stored or only purchases with the same articles and the same prices are stored. T_1 does not satisfy φ_2 , but T_1 does: the projections of all three pre-images are equivalent on $\llbracket \text{Article} \rrbracket$ and on $\llbracket \text{Price} \rrbracket$.

The constraint $\varphi_3 = v_{\text{Purchase}} : \llbracket \text{Customer} \rrbracket \vee \llbracket \text{Article,Price} \rrbracket$ expresses the fact that for any two distinct pre-images of $T(v_{\text{Purchase}})$ the projections on $\llbracket \text{Customer} \rrbracket$ must be equivalent or the projections on $\llbracket \text{Article,Price} \rrbracket$ must be equivalent. In other words, an XML data tree satisfies φ_3 when only purchases of the same customer are stored or only purchases with the same items are stored. Both T_1 and T_2 do not satisfy φ_3 .

The constraint $\varphi_4 = v_{\text{Purchase}} : \llbracket \text{Article,Price} \rrbracket \Rightarrow \llbracket \text{Discount} \rrbracket$ expresses the fact that for any two distinct pre-images of $T(v_{\text{Purchase}})$ if the projections on $\llbracket \text{Article,Price} \rrbracket$ are equivalent, then the projections on $\llbracket \text{Discount} \rrbracket$ must be equivalent, too. In other words, an XML data tree satisfies φ_4 when only purchases are stored in which the items uniquely determine the discount. Both T_1 and T_2 satisfy φ_4 . Notice that every pair of distinct pre-images of $T(v_{\text{Purchase}})$ in T_1 has different projections on $\llbracket \text{Article,Price} \rrbracket$. Hence, T_1 trivially satisfies φ_4 .

The constraint $\varphi_5 = v_{\text{Purchase}} : \llbracket \text{Discount} \rrbracket \Rightarrow (\llbracket \text{Customer} \rrbracket \vee \llbracket \text{Article,Price} \rrbracket)$ expresses the fact that for any two distinct pre-images of $T(v_{\text{Purchase}})$ whether the projections on $\llbracket \text{Discount} \rrbracket$ are equivalent, then the projections on $\llbracket \text{Customer} \rrbracket$ must be equivalent or the projections on $\llbracket \text{Article,Price} \rrbracket$ must be equivalent. In other words, an XML data tree satisfies φ_5 when only purchases are stored in which the discount determines the customer or determines the price. The data tree T_1 does not satisfy φ_5 because the second and third pre-image have equivalent projections on $\llbracket \text{Discount} \rrbracket$ but projections on $\llbracket \text{Customer} \rrbracket$ and $\llbracket \text{Article,Price} \rrbracket$ which are not equivalent. The data tree T_2 does indeed satisfy φ_5 .

4.3. Justification of the Definition

Note that the essential subgraphs in $\mathcal{B}(v)$ form the basis of any Boolean constraint expression. The results from the previous section provide a complete justification for this definition.

On one hand, omitting any essential subgraph results in an immediate loss of expressiveness of our constraints. For instance, without the essential subgraph $\llbracket \text{Article,Price} \rrbracket$ we cannot express the Boolean constraint

$$v_{\text{Purchase}} : \llbracket \text{Article,Price} \rrbracket \Rightarrow \llbracket \text{Discount} \rrbracket.$$

In fact, this constraint is different from

$$v_{\text{Purchase}} : (\llbracket \text{Article} \rrbracket \wedge \llbracket \text{Price} \rrbracket) \Rightarrow \llbracket \text{Discount} \rrbracket.$$

The XML tree T' in Figure 2 satisfies the first constraint, but does not satisfy the second constraint. Consequently, we would lose the ability to specify a desired constraint.

On the other hand, our definition does not have any overhead: permitting any v -subgraphs that are not essential subgraphs does not increase the expressiveness of Boolean constraints. For instance, the expression

$$v_{\text{Purchase}} : \llbracket \text{Customer, Article, Price} \rrbracket \Rightarrow \llbracket \text{Discount} \rrbracket$$

(whose definition of satisfaction is straightforward) is satisfied by the same XML data trees that satisfy the Boolean constraint

$$v_{\text{Purchase}} : (\llbracket \text{Customer} \rrbracket \wedge \llbracket \text{Article, Price} \rrbracket) \Rightarrow \llbracket \text{Discount} \rrbracket$$

since $\llbracket \text{Customer} \rrbracket$ and $\llbracket \text{Article, Price} \rrbracket$ are reconcilable v_{Purchase} -subgraphs.

4.4. The Implication Problem of Boolean Constraints

It is our goal for the rest of this paper to propose automated reasoning support for Boolean constraints. We say that Σ *implies* φ if and only if every XML data tree T' that satisfies all the Boolean constraints in Σ also satisfies the Boolean constraint φ . The *implication problem* is to decide, given any XML schema tree T any vertex $v \in V_T$ and any finite set $\Sigma \cup \{\varphi\}$ of Boolean constraints over v , whether Σ implies φ . The *implication problem on two pre-image data trees* is to decide, given any XML schema tree T any vertex $v \in V_T$ and any finite set $\Sigma \cup \{\varphi\}$ of Boolean constraints over v , whether every XML data tree T' that has precisely two pre-images of $T(v)$ and satisfies all the Boolean constraints in Σ also satisfies the Boolean constraint φ .

5. Reasoning about Boolean Constraints

In this section we will define the mapping between data dependencies and propositional formulae, and present the equivalence results. We start with some examples that illustrate the techniques that will be used in the proof arguments.

5.1. Propositional Logic

We repeat some basic notions from classical Boolean propositional logic [8]. Let \mathcal{V} denote a set of propositional variables. The set $\mathbb{F}_{\mathcal{V}}$ of propositional formulae over \mathcal{V} are recursively defined as follows:

- every propositional variable in \mathcal{V} is a formulae in $\mathbb{F}_{\mathcal{V}}$,
- if $\varphi \in \mathbb{F}_{\mathcal{V}}$, then $\neg\varphi \in \mathbb{F}_{\mathcal{V}}$,
- if $\psi_1, \psi_2 \in \mathbb{F}_{\mathcal{V}}$, then $(\psi_1 \wedge \psi_2) \in \mathbb{F}_{\mathcal{V}}$.

Let 0, 1 denote the propositional truth values *false* and *true*, respectively. A truth assignment over \mathcal{V} is a mapping $\theta : \mathcal{V} \rightarrow \{0, 1\}$ that assigns each variable in \mathcal{V} a truth value. A truth assignment over \mathcal{V} is extended to a function $\Theta : \mathbb{F}_{\mathcal{V}} \rightarrow \{0, 1\}$ that maps every formula φ in $\mathbb{F}_{\mathcal{V}}$ to its truth value $\Theta(\varphi)$ as follows:

- if $\varphi \in \mathcal{V}$, then $\Theta(\varphi) = \theta(\varphi)$,
- if $\varphi = \neg\psi$ for $\psi \in \mathbb{F}_{\mathcal{V}}$, then $\Theta(\varphi) = \begin{cases} 1 & , \text{ if } \Theta(\psi) = 0 \\ 0 & , \text{ otherwise} \end{cases}$,

- if $\varphi = \psi_1 \wedge \psi_2$ for $\psi_1, \psi_2 \in \mathbb{F}_{\mathcal{V}}$, then $\Theta(\varphi) = \begin{cases} 1 & , \text{ if } \Theta(\psi_1) = \Theta(\psi_2) = 1 \\ 0 & , \text{ otherwise} \end{cases}$.

Note that we will make frequent use of the abbreviations $(\varphi \vee \psi)$ for $\neg(\neg\varphi \wedge \neg\psi)$, and $(\varphi \Rightarrow \psi)$ for $(\neg\varphi \vee \psi)$, as it was the case with Boolean constraints.

We say that a truth assignment θ over \mathcal{V} *satisfies* the formalue φ in $\mathbb{F}_{\mathcal{V}}$, denoted by $\models_{\theta} \varphi$, if and only if $\Theta(\varphi) = 1$. We further say that a set Σ' of propositional formulae over \mathcal{V} *logically implies* the propositional formula φ' over \mathcal{V} if and only if every truth assignment that satisfies all the formulae in Σ' also satisfies the formula φ' .

5.2. Mapping between Constraints and Formulae

Let T be an XML schema tree, and $v \in V_T$ a vertex of T . Let $\phi : \mathcal{B}(v) \rightarrow \mathcal{V}$ denote a bijection between the essential subgraphs of v and the set \mathcal{V} of propositional variables. We will now extend this bijection to Boolean constraints over v and propositional formulae over \mathcal{V} .

We recursively define this mapping $\Phi : BC(v) \rightarrow \mathbb{F}_{\mathcal{V}}$ for Boolean constraints φ to their propositional formulae $\Phi(\varphi) = \varphi'$. If $\varphi = X \in \mathcal{B}(v)$ is an essential subgraph of v , then let $\varphi' = \phi(X)$. The rest of the mapping is straightforward:

- for $\varphi = \neg\psi$ we have $\varphi' = \neg\psi'$, and
- for $\varphi = (\psi_1 \wedge \psi_2)$ we have $\varphi' = (\psi'_1 \wedge \psi'_2)$.

If Σ is a set of Boolean constraints over v , then let $\Sigma' = \{\sigma' \mid \sigma \in \Sigma\}$ denote the corresponding set of propositional formulae over \mathcal{V} . Furthermore, the set

$$\Sigma'_v = \{\phi(X) \Rightarrow \phi(Y) \mid X, Y \in \mathcal{B}(v), X \text{ covers}^2 Y\}$$

denotes those formulae which encode the structure of (the essential subgraphs of) v .

Example 5 Consider the XML schema tree T in Figure 5, the vertex v_{Purchase} and the set Σ consisting of the two Boolean constraints

$$v_{\text{Purchase}} : \llbracket \text{Customer} \rrbracket \Rightarrow \llbracket \text{Discount} \rrbracket \text{ and } v_{\text{Purchase}} : \llbracket \text{Article, Price} \rrbracket \Rightarrow \llbracket \text{Discount} \rrbracket$$

and

$$\varphi = v_{\text{Purchase}} : \llbracket \text{Discount} \rrbracket \Rightarrow (\llbracket \text{Customer} \rrbracket \vee \llbracket \text{Article, Price} \rrbracket).$$

We define the mapping $\phi : \mathcal{B}(v_{\text{Purchase}}) \rightarrow \mathcal{V}$ as follows

- $\phi(\llbracket \text{Customer} \rrbracket) = V_1$,
- $\phi(\llbracket \text{Article} \rrbracket) = V_2$,
- $\phi(\llbracket \text{Price} \rrbracket) = V_3$,
- $\phi(\llbracket \text{Discount} \rrbracket) = V_4$, and
- $\phi(\llbracket \text{Article, Price} \rrbracket) = V_5$.

The resulting mappings into propositional formulae are then as follows:

² X covers Y iff $Y < X$ and for all $Z \in \mathcal{B}(v)$ with $Y \leq Z \leq X$ we have $Y = Z$ or $X = Z$, this is just the standard definition of a *cover relation* for posets, see [1]

- $\Sigma' = \{V_1 \Rightarrow V_4, V_5 \Rightarrow V_4\},$
- $\Sigma'_{v_{Purchase}} = \{V_5 \Rightarrow V_2, V_5 \Rightarrow V_3\},$ and
- $\varphi' = V_4 \Rightarrow (V_1 \vee V_5).$

□

Next, we will establish that these representations are, in fact, equivalent.

5.3. The Equivalence

We will now present the main result of this paper. They generalise results from the relational data model [10,20,21] where

- the underlying schema is flat,
- the join-irreducibles of the underlying schema form an anti-chain, and
- it is sufficient to consider join-irreducibles only.

In fact, the XML schema trees are not flat but have a complex structure, the essential subgraphs of a fixed vertex node do not form an anti-chain but a non-trivial partially ordered set, and it is not sufficient to consider unary subgraphs only. It is in this sense, that the following theorem generalises the results of [10,20,21].

Theorem 2 [Equivalence Theorem for Boolean Constraints in XML] *Let T be an XML schema tree, $v \in V_T$ a vertex of T , and $\Sigma \cup \{\varphi\}$ a set of Boolean constraints over v . Let Σ'_v denote the propositional formulae which encode the structure of v , and Σ' denote the corresponding set of propositional formulae for Σ . Then*

1. Σ implies φ ,
2. Σ implies φ on two pre-image data trees, and
3. $\Sigma' \cup \Sigma'_v$ logically implies φ'

are equivalent.

□

Example 6 *We continue Example 5. Evidently, the Boolean constraint φ is not implied by the set Σ of Boolean constraints over $v_{Purchase}$. In fact, the XML data tree T' in Figure 3 satisfies all the constraints in Σ , but does not satisfy φ .*

In logical terms, the formula φ' is not implied by the set Σ' of propositional formulae over \mathcal{V} . In fact, the truth assignment θ defined by $\theta(V_i) = 1$ if and only if $i \in \{2, 3, 4\}$ satisfies all the formulae in $\Sigma' \cup \Sigma'_{v_{Purchase}}$, but does not satisfy φ' . □

The crucial observation, illustrated by Example 6, is the strong correspondence between XML data trees T' that form a counterexample for the implication of Boolean constraints and truth assignments θ that form a counterexample for the implication of propositional formulae. In fact, the two pre-images of $T(v_{Purchase})$ in T' agree on projections to precisely those essential subgraphs X whose corresponding propositional variable V is assigned the truth value 1 by θ .

5.4. Proof Argument

We start off by showing the equivalence of 1. and 2. in Theorem 2. It is immediate that 1. implies 2. since every two pre-image data tree is also a data tree. The converse implication follows from Definition 4. Assume that 1. does not hold. That is, there is

some T -compatible XML data tree T' such that $\models_{T'} \sigma$ for all $\sigma \in \Sigma$, but not $\models_{T'} \varphi$. According to Definition 4 there are two distinct pre-images W_1, W_2 of v in T' such that not $\models_{\{W_1, W_2\}} \varphi$ holds. Since W_1, W_2 are pre-images of $T(v)$ in T' we must have $\models_{\{W_1, W_2\}} \sigma$ for all $\sigma \in \Sigma$. Consequently, Σ does not imply φ in the world of two pre-image data trees, i.e., not 2..

In order to complete the proof of Theorem 2 it remains to show the equivalence between 2. and 3. The key idea is to define truth assignments based on two pre-image data trees and vice versa. In fact, one interprets a variable as *true* precisely if the two pre-images agree on their projections to the corresponding essential subgraph of that variable.

Lemma 1 *Let T be an XML schema tree, $v \in V_T$ a vertex of T , T' a T -compatible XML data tree such that W_1, W_2 are distinct pre-images of $T(v)$ in T' , and φ be a Boolean constraint over v . Then $\models_{\{W_1, W_2\}} \varphi$ if and only if $\models_{\theta_{\{W_1, W_2\}}} \varphi'$ where*

$$\theta_{\{W_1, W_2\}}(V) = \begin{cases} 1, & \text{if } W_1|_{\phi^{-1}(V)} \text{ is equivalent to } W_2|_{\phi^{-1}(V)} \\ 0, & \text{else} \end{cases}$$

for all $V \in \phi(\mathcal{B}(v))$. □

The proof of Lemma 1 is done by induction on the structure of φ . Let $\varphi = X \in \mathcal{B}(v)$. We then have $\models_{\{W_1, W_2\}} X$ if and only if $W_1|_X$ is equivalent to $W_2|_X$ by Definition 4. The last condition, however, is equivalent to $\theta_{\{W_1, W_2\}}(\phi(X)) = 1$. This shows the start of the induction. The induction steps are a straightforward application of Definition 4.

The following observation enables one to prove the equivalence between 2. and 3. of Theorem 2. We call a set \mathcal{X} of v -subgraphs an *ideal* if it has the following properties:

- $\llbracket \emptyset \rrbracket \in \mathcal{X}$,
- \mathcal{X} is closed under the v -subgraph union of reconcilable elements, i.e., if $X, Y \in \mathcal{X}$ and X, Y are reconcilable, then $X \sqcup Y \in \mathcal{X}$, and
- \mathcal{X} is closed downwards, i.e., if $X \in \mathcal{X}$ and Y is a v -subgraph of X , then $Y \in \mathcal{X}$.

One can now generate for any ideal \mathcal{X} a T -compatible XML data tree T' with two pre-images W_1 and W_2 of $T(v)$ in T' such that for all v -subgraphs $X \in \text{Sub}_T(v)$ we have that W_1 and W_2 have equivalent projections on X precisely if $X \in \mathcal{X}$ holds. We will illustrate the general construction of this key observation by an example.

Let v be the node v_{purchase} in the XML schema tree on the left of Figure 7. Let \mathcal{X} be the set consisting of its three unary v -subgraphs, see Figure 7 and the empty v -subgraph $\llbracket \emptyset \rrbracket$. Then \mathcal{X} is indeed an ideal (no pair of non-empty v -subgraphs is reconcilable) and we should be able to generate two pre-images with the desired property.

In a first step we determine all those minimal v -subgraphs on which the pre-images should differ. In this case, these are exactly the binary v -subgraphs, i.e. those with two leaves, see Figure 7. Now we choose different strings, assign these strings to copies of the binary v -subgraphs and generate the two pre-images by merging the copies appropriately. We omit the technical details of this construction. However, the technique is based on the observation that for a fixed vertex v the set of v -subgraphs carries the structure of a Boolean algebra. In order to make the XML data trees as realistic as possible we choose

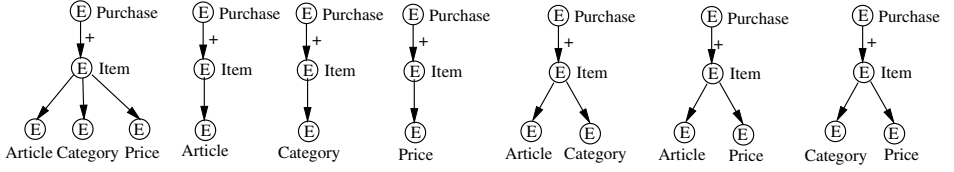
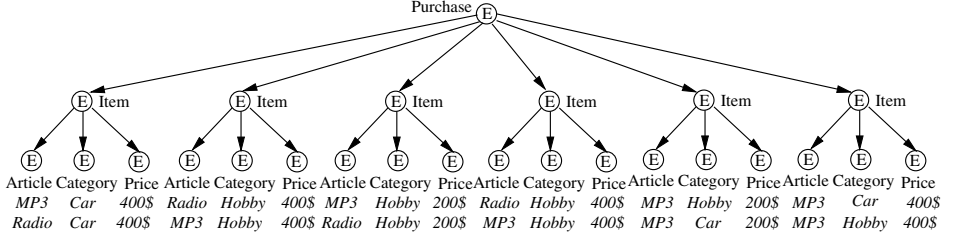


Figure 7. XML schema graph, its unary and binary subgraphs

Figure 8. Pre-images W_1 and W_2 of $T(v_{purchase})$

reasonable strings for *Article*, *Category* and *Price*. In this example we choose the strings *MP3* and *Radio* for *Article*, *Hobby* and *Car* for *Category*, and *400\$* and *200\$* for *Price*.

The pre-images W_1 and W_2 of $T(v_{purchase})$ are illustrated in Figure 8 where the data values of W_1 (W_2) appear in the top (bottom) row.

One can easily verify that for all $v_{purchase}$ -subgraphs $X \in Sub_T(v_{purchase})$ the projections $W_1|_X$ and $W_2|_X$ are equivalent precisely when $X \in \mathcal{X}$ holds.

We are now prepared to show the equivalence between 2. and 3. Suppose 2. does not hold. Then there is some T -compatible XML data tree T' with exactly two distinct pre-images W_1, W_2 of $T(v)$ in T' such that $\models_{\{W_1, W_2\}} \sigma$ for all $\sigma \in \Sigma$, but not $\models_{\{W_1, W_2\}} \varphi$. According to Lemma 1 we know that $\models_{\theta_{\{W_1, W_2\}}} \sigma'$ for all $\sigma' \in \Sigma'$ and not $\models_{\theta_{\{W_1, W_2\}}} \varphi'$.

We will show that $\models_{\theta_{\{W_1, W_2\}}} \Sigma'_v$. Let $\phi(X) \rightarrow \phi(Y) \in \Sigma'_v$. According to the definition of Σ'_v we have that Y is a v -subgraph of X . Suppose that $\theta_{\{W_1, W_2\}}(\phi(X)) = 1$. Then $W_1|_X$ and $W_2|_X$ are equivalent according to the definition of the truth assignment $\theta_{\{W_1, W_2\}}$. Since Y is a v -subgraph of X it follows that $W_1|_Y$ and $W_2|_Y$ are equivalent as well. This means, however, that $\theta_{\{W_1, W_2\}}(\phi(Y)) = 1$, too.

Consequently, φ' is not logically implied by $\Sigma' \cup \Sigma'_v$ as witnessed by $\theta_{\{W_1, W_2\}}$. That means 3. does not hold and it remains to show that 2. implies 3.

Suppose 3. does not hold. Then there is some truth assignment θ which makes every formula in $\Sigma' \cup \Sigma'_v$ true, but makes φ' false. It is now sufficient to find some T -compatible XML data tree T' with exactly two distinct pre-images W_1, W_2 of $T(v)$ in T' such that $\theta = \theta_{\{W_1, W_2\}}$. In this case, Lemma 1 shows that $\models_{T'} \sigma$ for all $\sigma \in \Sigma$ and not $\models_{T'} \varphi$, i.e., 2. does not hold.

Let $\mathcal{X} = \{X \in Sub_T(v) \mid \forall Y \in \mathcal{B}(v). X \sqcap Y = [\emptyset] \text{ or } \theta(\phi(X \sqcap Y)) = 1\} \cup \{[\emptyset]\}$. The set \mathcal{X} is an ideal:

1. $\mathcal{X} \neq \emptyset$ since $[\emptyset] \in \mathcal{X}$.
2. \mathcal{X} is closed downwards with respect to v -subgraphs: let $X \in \mathcal{X}$ and Y be a v -subgraph of X . Since $X \in \mathcal{X}$ we have $\theta(\phi(X \sqcap Z)) = 1$ or $X \sqcap Z = [\emptyset]$ for all $Z \in \mathcal{B}(v)$. As $\models_{\theta} \Sigma'_v$ we also have $\models_{\theta} \phi(X \sqcap Z) \rightarrow \phi(Y \sqcap Z)$. Consequently,

$\theta(\phi(Y \sqcap Z)) = 1$ or $Y \sqcap Z = \llbracket \emptyset \rrbracket$ holds for all $Z \in \mathcal{B}(v)$ which means that $Y \in \mathcal{X}$ holds as well.

3. \mathcal{X} is closed under the v -subgraph union of reconcilable elements: let $X, Y \in \mathcal{X}$ be reconcilable. Since X, Y are reconcilable it follows $\forall Z \in \mathcal{B}(v)$ that $X \sqcap Z$ is a v -subgraph of $Y \sqcap Z$ or vice versa. We know that $\forall Z \in \mathcal{B}(v)$ we have $(\theta(\phi(X \sqcap Z)) = 1 \text{ or } X \sqcap Z = \llbracket \emptyset \rrbracket)$ and $(\theta(\phi(Y \sqcap Z)) = 1 \text{ or } Y \sqcap Z = \llbracket \emptyset \rrbracket)$. Consequently, it follows that $\forall Z \in \mathcal{B}(v)$ we have $\theta(\phi((X \sqcup Y) \sqcap Z)) = 1$ or $(X \sqcup Y) \sqcap Z = \llbracket \emptyset \rrbracket$ holds. This implies that $X \sqcup Y \in \mathcal{X}$, too.

According to these properties there is a T -compatible XML data tree T' with two pre-images W_1 and W_2 of $T(v)$ in T' such that for all v -subgraphs $X \in \text{Sub}_T(v)$ we have that W_1 and W_2 have equivalent projections on X precisely if $X \in \mathcal{X}$ holds. For all $X \in \mathcal{B}(v)$ we conclude,

$$\begin{aligned} \theta_{\{W_1, W_2\}}(\phi(X)) = \text{true} & \Leftrightarrow W_1|_X \text{ is equivalent to } W_2|_X \\ & \Leftrightarrow X \in \mathcal{X} \\ & \Leftrightarrow \theta(\phi(X)) = \text{true} \end{aligned}$$

This completes the proof of Theorem 2.

6. Some Subclasses of Boolean Constraints

The following result is a consequence of Theorem 2 and the *NP*-completeness of the *satisfiability problem* for propositional clauses [6,19].

Corollary 1 *Let T be an XML schema tree, and $v \in V_T$ a vertex in T . The implication problem of Boolean constraints over v is coNP-complete to decide.* \square

It is therefore interesting to ask what common subclasses of Boolean constraints still have an associated implication problem that can be decided efficiently.

Let T be an XML schema tree, and $v \in V_T$ a vertex in T . A *functional dependency* on v is an expression $v : \mathcal{X} \rightarrow \mathcal{Y}$ where \mathcal{X}, \mathcal{Y} are sets of essential subgraphs on v . An XML data tree T' that is compatible with T is said to satisfy the functional dependency $v : \mathcal{X} \rightarrow \mathcal{Y}$ if and only if for all pre-images W_1, W_2 of $T(v)$ in T' the following holds: if $W_1|_X$ and $W_2|_X$ are equivalent for all $X \in \mathcal{X}$, then $W_1|_Y$ and $W_2|_Y$ are equivalent for all $Y \in \mathcal{Y}$. It is immediate that a functional dependency $v : \{X_1, \dots, X_n\} \rightarrow \{Y_1, \dots, Y_m\}$ is satisfied by T' if and only if T' satisfies the Boolean constraint $v : (X_1 \wedge \dots \wedge X_n) \Rightarrow (Y_1 \wedge \dots \wedge Y_m)$. Hence, functional dependencies are a special case of Boolean constraints. In fact, they correspond to Horn clauses in propositional logic [14], but the implication of Horn clauses is decidable in time linear in the total number of variables [7].

Corollary 2 *Let T be an XML schema tree, and $v \in V_T$ a vertex in T . The implication problem of functional dependencies over v can be decided in time linear in the total number of essential subgraphs of v occurring in the input instance.* \square

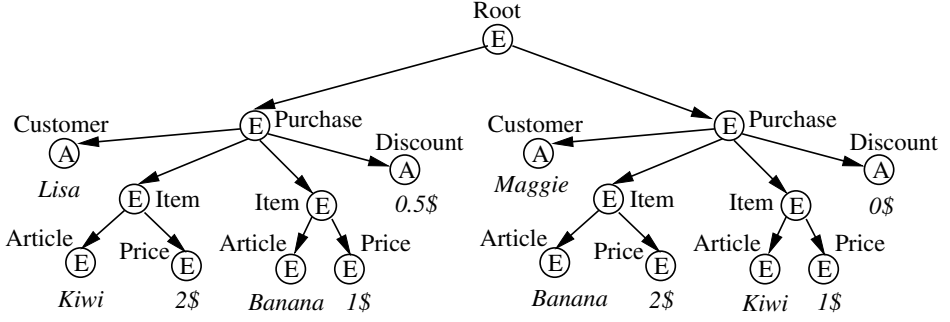


Figure 9. An XML data tree showing that $v_{\text{Purchase}} : \llbracket \text{Article} \rrbracket, \llbracket \text{Price} \rrbracket \rightarrow \llbracket \text{Discount} \rrbracket$ is not implied by $v_{\text{Purchase}} : \llbracket \text{Article}, \text{Price} \rrbracket \rightarrow \llbracket \text{Discount} \rrbracket$

Example 7 Consider the XML schema tree T from Figure 5. The XFD

$$v_{\text{Purchase}} : \llbracket \text{Article} \rrbracket, \llbracket \text{Price} \rrbracket \rightarrow \llbracket \text{Discount} \rrbracket$$

is not implied by the XFD

$$v_{\text{Purchase}} : \llbracket \text{Article}, \text{Price} \rrbracket \rightarrow \llbracket \text{Discount} \rrbracket.$$

A counter-example data tree is illustrated in Figure 9. Let ϕ denote the mapping of essential subgraphs in $\mathcal{B}(v_{\text{Purchase}})$ to propositional variables from Example 5. Then we obtain

$$\Sigma' = \{\neg V_5 \vee V_4\}, \Sigma'_{v_{\text{Purchase}}} = \{\neg V_5 \vee V_2, \neg V_5 \vee V_3\} \text{ and } \varphi' = \neg V_2 \vee \neg V_3 \vee V_4.$$

The truth assignment θ with $\theta(V_i) = 1$ if and only if $i \in \{2, 3\}$ shows that φ' is not logically implied by $\Sigma' \cup \Sigma'_{v_{\text{Purchase}}}$. Note that θ assigns 1 to precisely those variables on whose corresponding essential subgraphs the two pre-images in T' are equivalent. \square

A degenerated multivalued dependency on v is an expression $v : \mathcal{X} \rightarrow \mathcal{Y} \mid \mathcal{Z}$ where $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ are sets of essential subgraphs on v such that $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z} = \mathcal{B}(v)$. An XML data tree T' that is compatible with T is said to satisfy the degenerated multivalued dependency $v : \mathcal{X} \rightarrow \mathcal{Y} \mid \mathcal{Z}$ if and only if for all pre-images W_1, W_2 of $T(v)$ in T' the following holds: if $W_1|_{\mathcal{X}}$ and $W_2|_{\mathcal{X}}$ are equivalent for all $X \in \mathcal{X}$, then $W_1|_{\mathcal{Y}}$ and $W_2|_{\mathcal{Y}}$ are equivalent for all $Y \in \mathcal{Y}$ or $W_1|_{\mathcal{Z}}$ and $W_2|_{\mathcal{Z}}$ are equivalent for all $Z \in \mathcal{Z}$. It is immediate that the degenerated multivalued dependency $v : \{X_1, \dots, X_n\} \rightarrow \{Y_1, \dots, Y_m\} \mid \{Z_1, \dots, Z_l\}$ is satisfied by T' if and only if T' satisfies the Boolean constraint $v : (X_1 \wedge \dots \wedge X_n) \Rightarrow ((Y_1 \wedge \dots \wedge Y_m) \vee (Z_1 \wedge \dots \wedge Z_l))$. Hence, degenerated multivalued dependencies are a special case of Boolean constraints. As the implication of degenerated multivalued dependencies corresponds to the implication of multivalued dependencies [20,21], and their implication is decidable in time $\mathcal{O}(n \cdot \log n)$ where n denotes the total number of attributes occurring in the input instance [12], we obtain the following result.

Corollary 3 *Let T be an XML schema tree, and $v \in V_T$ a vertex in T . The implication problem of degenerated multivalued dependencies over v can be decided in time $\mathcal{O}(n \cdot \log n)$ where n denotes the total number of essential subgraphs of v occurring in the input instance.* \square

A 2-Literal constraint over v is a Boolean constraint over v of the following form $v : L_1 \vee L_2$ with $L_j = X \in \mathcal{B}(v)$ or $L_j = \neg X$ for some $X \in \mathcal{B}(v)$ for all $j = 1, 2$. The following result follows immediately from the decidability of 2-SAT in linear time [4,9].

Corollary 4 *Let T be an XML schema tree, and $v \in V_T$ a vertex in T . The implication problem of 2-Literal constraints over v is decidable in time linear in the total number of essential subgraphs of v occurring in the input instance.* \square

7. Conclusion and Future Work

We have introduced the class of Boolean constraints into XML. These extend the notion of Boolean dependencies from relational databases, and are based on homomorphisms between XML schema trees and XML data trees. We have justified the definition of Boolean constraints by demonstrating which subgraphs of a fixed vertex v in any XML schema tree T determine pre-images of $T(v)$ up to equivalence. Boolean constraints allow us to express many properties of XML data that cannot be expressed by other classes of XML constraints. Moreover, we have shown that the implication problem of Boolean constraints is equivalent to the implication problem of propositional formulae. Hence, reasoning about Boolean constraints is well founded, and off-the-shelf SAT solvers from artificial intelligence research can be directly applied. While the implication problem of Boolean constraints is *coNP*-complete to decide in general, we have identified several subclasses that can be reasoned about efficiently.

In future work, one may study the interactions of Boolean constraints over different vertices v . While this increases the expressiveness it is also likely to increase the time-complexity of the associated decision problems. Our definition of Boolean constraints assumes a multiset semantics among the labels of v -descendants. While this is the most challenging semantics there are also application domains that favour a set or list semantics.

Similar to the first kind of XFDs our Boolean constraints also cause redundancy among XML data [2,29]. It is an interesting question how well-designed XML documents can be characterised. One is tempted to define normal forms that extend the well-known counterparts such as 3NF or BCNF from relational databases. Recently, an information-theoretic analysis has been applied to normal forms [3,17].

References

- [1] I. Anderson. *Combinatorics of finite sets*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1987.
- [2] M. Arenas and L. Libkin. A normal form for XML documents. *ToDS*, 29(1):195–232, 2004.
- [3] M. Arenas and L. Libkin. An information-theoretic approach to normal forms for relational and XML data. *J. ACM*, 52(2):246–283, 2005.

- [4] B. Aspvall, M. Plass, and R. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *IPL*, 8(3):121–123, 1979.
- [5] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (XML) 1.0 (third edition) W3C recommendation, feb. 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/>.
- [6] S. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158, 1971.
- [7] W. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Logic Programming*, 1(3):267–284, 1984.
- [8] H. Enderton. *A mathematical introduction to logic*. Academic Press, 2001.
- [9] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multi-commodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.
- [10] R. Fagin. Functional dependencies in a relational data base and propositional logic. *IBM Journal of Research and Development*, 21(6):543–544, 1977.
- [11] W. Fan. XML constraints. In *DEXA Workshops*, pages 805–809, 2005.
- [12] Z. Galil. An almost linear-time algorithm for computing a dependency basis in a relational database. *Journal of the ACM*, 29(1):96–102, 1982.
- [13] S. Hartmann and S. Link. More functional dependencies for XML. In *AdBIS*, number 2798 in LNCS, pages 355–369. Springer, 2003.
- [14] S. Hartmann, S. Link, and T. Trinh. Efficient reasoning about XFDs with Pre-Image semantics. In *DASFAA*, number 4443 in LNCS, pages 1070–1074. Springer, 2007.
- [15] S. Hartmann and T. Trinh. Axiomatising functional dependencies for XML with frequencies. In *FoIKS*, number 3861 in LNCS, pages 159–178. Springer, 2006.
- [16] D. Jungnickel. *Graphs, Networks and algorithms*. Springer, 1999.
- [17] S. Kolahi and L. Libkin. On redundancy vs dependency preservation in normalization: An information-theoretic study of 3NF. In *PoDS*, pages 114–123, 2006.
- [18] M. Lee, T. Ling, and W. Low. Designing functional dependencies for XML. In *EDBT*, number 2287 in LNCS, pages 124–141. Springer, 2002.
- [19] L. Levin. Universal sorting problems. *Problems of Information Transmission*, 9(3):265–266, 1973.
- [20] Y. Sagiv, C. Delobel, D. S. Parker Jr., and R. Fagin. An equivalence between relational database dependencies and a fragment of propositional logic. *Journal of the ACM*, 28(3):435–453, 1981.
- [21] Y. Sagiv, C. Delobel, D. S. Parker Jr., and R. Fagin. Correction to "An equivalence between relational database dependencies and a fragment of propositional logic". *Journal of the ACM*, 34(4):1016–1018, 1987.
- [22] Z. Tari, J. Stokes, and S. Spaccapietra. Object normal forms and dependency constraints for object-oriented schemata. *ToDS*, 22:513–569, 1997.
- [23] B. Thalheim. *Dependencies in Relational Databases*. Teubner-Verlag, 1991.
- [24] M. Vincent and J. Liu. Completeness and decidability properties for functional dependencies in XML. CoRR cs.DB/0301017, 2003.
- [25] M. Vincent, J. Liu, and C. Liu. Strong functional dependencies and their application to normal forms in XML. *ToDS*, 29(3):445–462, 2004.
- [26] J. Wang. A comparative study of functional dependencies for XML. In *APWeb*, number 3399 in LNCS, pages 308–319. Springer, 2005.
- [27] J. Wang and R. Topor. Removing XML data redundancies using functional and equality-generating dependencies. In *ADC*, number 39 in CRPIT, pages 65–74, 2005.
- [28] G. Weddell. Reasoning about functional dependencies generalized for semantic data models. *ToDS*, 17(1):32–64, 1992.
- [29] X. Wu, T. Ling, S. Lee, M. Lee, and G. Dobbie. NF-SS: A normal form for semistructured schema. In *ER Workshops*, number 2465 in LNCS, pages 292–305. Springer, 2001.
- [30] P. Yan and T. Lv. Functional dependencies in XML documents. In *APWeb Workshops*, number 3842 in LNCS, pages 29–37. Springer, 2006.

An Image-Query Creation Method for Representing Impression by Color-based Combination of Multiple Images

Shiori SASAKI ^a, Yoshiko ITABASHI ^b, Yasushi KIYOKI ^c and Xing CHEN ^d

^a *Graduate School of Media and Governance, Keio University*

^b *Keio Research Institute at SFC*

^c *Faculty of Environment and Information Studies, Keio University
5322 Endo, Fujisawa-shi, Kanagawa, JAPAN*

{sashiori, itabasiy, kiyoki}@mdbl.sfc.keio.ac.jp, kiyoki@sfc.keio.ac.jp

^d *Department of Information & Computer Sciences, Kanagawa Institute of Technology
1030 Shimo-Ogino, Atsugi, Kanagawa, JAPAN
chen@ic.kanagawa-it.ac.jp*

Abstract. This paper presents a dynamic image-query creation and metadata extraction method with semantic correlation computation between color-combinations and impressions of multiple image data. The main features of our method are (1) to create an image-query which reflects user's intention dynamically according to the color-based combinations of images with common features selected by a user as context, (2) to extract appropriate impression by each image collection which cannot be easily extracted from a single image, (3) to provide users an image retrieval environment reflecting historical and cultural semantics and impression of color especially for cultural properties, and (4) to enable an image retrieval environment for the collection of images by time, culture, author e.t.c.. The queries are created by the combination of multiple image sets and operations, which are intersection, accumulation, average, difference of color elements of sample images. First, a set of multiple images with common features is set as sample data for a query creation. Second, color histograms are extracted from the image sets for creating feature vector of a query. Third, the correlations between an image-query vector and target image vectors are calculated on a space which represents the relationship between color and the impression according to historical and cultural semantics of color. This image-query creation method representing impression of color makes it possible to expand the range of image retrieval for a large number of image data of cultural property in digital archives, such as electronic library and electronic museum, automatically.

1. Introduction

In this paper, we propose a dynamic image-query creation and metadata extraction method with semantic correlation computation between color-combinations and impressions of multiple image data. The created image-queries are representing impressions of colors by any specific author, time, culture, which is selected as a user's context. We also present a method to highlight minor color used in the image collection manufactured in a specific style, format or background and extract appropriate impressions which cannot be easily extracted from single image data.

By the diffusion of advances in multimedia technology, a large number of various type of multimedia data, such as images, audio and motion pictures, are created and distributed widely. To retrieve these data efficiently and appropriately, automatic metadata extraction methods for multimedia data, especially for image data, are studied extensively [1] - [10]. Based on the premise that features of images are able to be represented by the colors and coloration used in each image, a lot of metadata extraction methods by using color information for image data have been proposed [1][2][5][6][7]. However, the problem of “semantic gap” [2] is not solved sufficiently. That is, images with high similarities to queries do not necessarily represent or match to user’s intention and context in semantics. On the other hands, among the methods to extract features from color information of images, there are methods to extract several color combination coloration as “impressions” directly from each image [6][7] or extract weighted words as impressions according to all the color which used in each image [8]. These methods are applicable especially to images which have different compositions and styles. However, in cases where we apply these methods to images which created by only different composition or format but also different historical or cultural background, we have difficulty in extracting appropriate metadata because of different semantics of color [11][12][13].

To fill a semantic gap between image features and user’s context and use these varieties of historical and cultural semantics of color positively, we propose an image-query creation method for representing impression by color-based combinations of multiple images and apply the method to an image retrieval system especially for cultural properties. First, a set of multiple images with common features is set as sample data for a query creation. Second, color histograms are extracted from the image sets for creating a feature vector of a query. Third, the correlations between an image-query vector and target image vectors are calculated on a space which represents the relationships between colors and the impressions according to historical and cultural semantics of color. To verify the feasibility and effectiveness of this image-query creation and metadata extraction method, we performed qualitative experiments with the image data, especially with Japanese cultural properties in digital archives.

2. An Image-Query Creation Method by Color-based Combination of Multiple Images

2.1 Features of the method

In this section, we present the main features of our image-query creation method and image retrieval environment especially for cultural properties reflecting historical and cultural semantics and impression of color.

This image-query creation method is based on the assumption that impression of color is very different according to historical and cultural background by the following observation results. (a) Some kind of patterns in use of color can be seen in art works by painter, school, collection and a series of works in the same style. (b) Some kind of patterns in coloration can be seen by age when art works painted especially for cultural property because of the historical/cultural background.

Based on these observation results, we propose an image-query creation by extracting color frequency of appearance in a set of images as features representing impression or “impression-context” of the image set. This method also based on the assumption that the dominant colors in an image collection represent an impression of the whole collection, and the characteristic colors in each image or each collection that are not used in other images or other collection determine an impression of each image or each collection.

The main features of our method are (1) to create an image-query which reflects user's intention dynamically according to the color-based combinations of images with common features selected by a user as context, (2) to extract appropriate impression by each image collection which cannot be easily extracted from a single image, (3) to provide users an image retrieval environment reflecting historical and cultural semantics and impression of color especially for cultural properties, and (4) to enable an image retrieval environment for the collection of images by time, culture, author etc.

The image-queries are created by the combination of multiple image sets and the following operations, after calculating the area ratio of color distribution of each sample image data: Operation 1 to create vector $Q_{average}$ by the averagely-used color from all the sample images in a set, Operation 2 to create vector $Q_{accumulation}$ by all the color from all the sample images in a set, Operation 3 to create vector $Q_{intersection}$ by the commonly-used color from all the sample images in a set, Operation 4 to create vector $Q_{difference}$ by the colors less frequently used in a single sample image compared with any other sample images in a set, or the colors less frequently used in a set of sample images compared with other sets of sample images.

2.2 Query creation method

For given n sample images ($s_1^l, s_2^l, \dots, s_n^l$: l is a set identifier) which represent p sets of images (l_1, l_2, \dots, l_p), color distribution information of each image data is extracted. From the extracted color distribution information of each sample image, color histograms are generated by calculating the area ratio of m basic colors (c_1, c_2, \dots, c_m) as number of pixels/all the pixels of the image.

The relationships between each sample image and sample image sets are not necessarily prepared as mutually exclusive relationships. It is possible that an image is included in multiple image sets, and a set of images is included in other sets of images.

The area ratio of m basic colors for each sample image representing each image set is defined as color-image set vector $s_k^l = \{s_{k1}^l, s_{k2}^l, \dots, s_{km}^l$: l is a set identifier $\}$, and n by m matrix consisting of the color-image set vectors as row vectors is defined as color-image set matrix C as shown in Figure 1. In other words, the color-image set matrix C represents the color features of each image set as numerical values ($q_{11}, q_{12}, \dots, q_{nm}$) of color histograms of n sample image data.

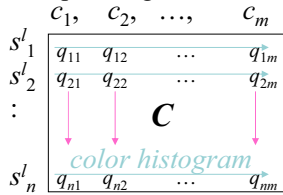


Figure 1: Color-Image set Matrix C

In this image-query creation method, a query vector is constructed from the color features of image sets by using the column vectors of the color-image set matrix C .

Operation 1: Average

To create a query vector $Q_{average}$, the averaged values of color elements are calculated from all the colors used in a set of sample images.

For the values of generated color histograms ($q_{11}, q_{12}, \dots, q_{nm}$) from each sample image data ($s_1^l, s_2^l, \dots, s_n^l$), the average values are calculated by every color elements (c_1, c_2, \dots, c_m), that is, by each column vector of matrix C . The calculated average values are constructed as elements of a query vector $Q_{average}$.

$$Q_{average} = \frac{1}{n} \left(\sum_{j=1}^n q_{j1}, \sum_{j=1}^n q_{j2}, \dots, \sum_{j=1}^n q_{jm} \right) \quad (1)$$

Operation 2: Accumulation

To create a query vector $Q_{accumulation}$, the maximum values of color elements are calculated from all the colors used in a set of sample images.

For the values of generated color histograms ($q_{11}, q_{12}, \dots, q_{nm}$) from each sample image data (s'_1, s'_2, \dots, s'_n), the maximum values are calculated by every color elements (c_1, c_2, \dots, c_m), that is, by each column vector of matrix C . The calculated maximum values are constructed as elements of a query vector $Q_{accumulation}$.

$$Q_{accumulation} = (\max(q_{11}, q_{21}, \dots, q_{n1}), \max(q_{12}, q_{22}, \dots, q_{n2}), \dots, \max(q_{1m}, q_{2m}, \dots, q_{nm})) \quad (2)$$

Operation 3: Intersection

To create a query vector $Q_{intersection}$, the minimum values of color elements are calculated from all the colors used in a set of sample images.

For the values of generated color histograms ($q_{11}, q_{12}, \dots, q_{nm}$) from each sample image data (s'_1, s'_2, \dots, s'_n), the minimum values are calculated by every color elements (c_1, c_2, \dots, c_m), that is, by each column vector of matrix C . The calculated minimum values are constructed as elements of a query vector $Q_{intersection}$.

$$Q_{intersection} = (\min(q_{11}, q_{21}, \dots, q_{n1}), \min(q_{12}, q_{22}, \dots, q_{n2}), \dots, \min(q_{1m}, q_{2m}, \dots, q_{nm})) \quad (3)$$

Operation 4: Difference

To create a query vector $Q_{difference}$, the difference values of color elements are calculated from the colors only used in a single sample image compared with any other sample images in a set, or from the colors only used in a set of sample images compared with other sets of sample images.

For the values of generated color histograms ($q_{11}, q_{12}, \dots, q_{nm}$) from each sample image data (s'_1, s'_2, \dots, s'_n), the difference values of characteristic color elements of single image data are calculated with the other values of color elements of the other data.

When the element of $Q_{difference}(s'_k)$ is defined as v_{kj} ,

$$\text{If } q_{kj} < \max(q_{k1}, q_{k2}, \dots, q_{km}) \rightarrow \text{then } v_{kj} = 0 \text{ else } v_{kj} = q_{kj} \quad (4)$$

Or, when the difference values of color elements of a set of images calculated with the other values of color elements of the other image sets, the $Q_{difference}$ is calculated simply as follows.

$$\text{If } q_{kj} < Q_{averagej} \rightarrow \text{then } v_{kj} = 0 \text{ else } v_{kj} = q_{kj} \quad (5)$$

3. Implementation of Image Retrieval with the Created Image-Queries Representing Impression

3.1 Construction of a Retrieval Space for Images using Semantics of Color

By using research results of color science, color psychology or historical study of color, a semantic retrieval space which represents the relationships between color and impression are created.

For r impression words (e_1, e_2, \dots, e_r), each word is characterized by m basic colors (c_1, c_2, \dots, c_m). The r impression words are given in the form of an r by m matrix M . By using this matrix M , the orthogonal space is computed as the color-impression space $C-IS$ for image retrieval based on a mathematic model of meaning [5][6][7]. The eigenvalue decomposition of the correlation matrix of M is executed and the eigenvectors are normalized. The color-impression space $C-IS$ is defined as the span of the eigenvectors which correspond to nonzero eigenvalues.

3.2 Mapping Retrieval Target Images onto the Retrieval Space

As the feature quantity of a retrieval target image H , color distribution information of target image data are extracted. From the extracted color distribution information of target image H , color histograms are generated by calculating the area ratio of m basic colors (c_1, c_2, \dots, c_m) as number of pixels/all the pixels of the image. The generated color histogram is defined as a feature vector of target image $h = \{h_1, h_2, \dots, h_m\}$.

For a series of image collection such as picture scrolls and image sets with a story, it is possible to detect the background color by statistical methods. For an image collection which is painted in a certain time and style, it is also possible to weight the characteristic colors that do not appear frequently in the other images in the collection but appear in an image.

We define the weighting to the color histogram for those target image collections as *IIF* (Inverse Image Frequency), here. The weighting process is as follows.

STEP 1: Color histograms for all the images in a collection are generated, and the number of images which include a certain area ratio of each color (c_1, c_2, \dots, c_m) more than a threshold ε is calculated.

STEP 2: IIF values for weighting are calculated for each color as follows.

$$\begin{aligned} n(c) \neq 0 \quad iif(c) &= \log \left(\frac{N}{n(c)} \right) & \varepsilon : \text{Threshold of area ratio of color} \\ & & N : \text{The number of target images} \\ n(c) = 0 \quad iif(c) &= 0 & n(c): \text{The number of images} \\ & & \text{which } C > \varepsilon \end{aligned}$$

STEP 3: Weighted color histogram h' is generated for each image by the calculated coefficient value for weighting.

$$h'_i = iif(c_i) \cdot h_i$$

3.3 Correlation Calculation between Target Images and Queries

The generated target vectors h and h' are mapped onto the color-impression space $C-IS$. Then the correlation values between the target vectors and a query vector Q created according to a user's context are calculated on the $C-IS$. The correlated values are calculated by the Semantic Associative Search [5][6][7] or Cosine Measure.












4. Experiment

To verify the feasibility and effectiveness of our method, we performed qualitative experiments with the image data of cultural property in digital archives.

As an experimental environment, we created the color-impression space $C-IS$ using 120 chromatic colors and 10 monochrome colors defined in the "Color Image Scale" [14] based on the Munsell color system. We used 180 words, which are also defined as cognitive scale of color in the Color Image Scale, as impression words in the process of construction of $C-IS$. To generate color histograms of sample images for queries and retrieval target images, we used 130 colors, the same number of colors used in $C-IS$. We converted RGB values to HSV values per pixel of each image, clustered them into the closest color of 130 colors, and calculated the percentage of each color in all pixels of the image.

4.1 Experiment 1:

First, as a pilot study of image retrieval, we selected 50 sample images of Alphonse Mucha's art works for a query creation as an example of Western Art of the 19th Century. Second, we selected 36 images of Mucha's art works by Google Image

Query		Target Image	Histogram	Correlation		Target Image	Histogram	Correlation
 Intersection	1	mucha22.jpg		0.603851	6	mucha32.jpg		0.530148
	2	mucha33.jpg		0.576515	7	mucha17.jpg		0.494901
	3	mucha35.jpg		0.545627	8	mucha24.jpg		0.468386
	4	mucha12.jpg		0.533434	9	mucha16.jpg		0.452379
	5	mucha13.jpg		0.530234	10	mucha5.jpg		0.451566













Highly-correlated Impression words & the Histograms	
	sturdy, classic, pastoral, nostalgic, diligent, dignified, artistic & tasteful, chic, formal

Figure 2: Retrieval results by the query “Intersection” of a set of sample images painted by the same author

Search [17] differently from the sample images as retrieval targets. Third, we set several queries created by our method, and compared the retrieval results.

Figure 2 shows the retrieval results by the query of “Intersection” of 50 sample images of Mucha’s art works. The color histogram of the query shows that the commonly-used colors in Mucha’s art works are dark brown and grayish blue. The original histograms of retrieved target images show that retrieval results have the same impression and use typical coloration of Mucha’s art works. The highly-correlated impression words to the retrieved target images were “sturdy”, “classic”, “pastoral”, “nostalgic”, “diligent” and so on.

On the other hands, Figure 3 shows the retrieval results by the query “Difference” of “muchal8.jpg” from other 49 sample images. The histogram of the query shows that the characteristic colors used in “muchl8.jpg” compared to the other Mucha’s art works are light beige and flesh color. The retrieval results show that the histograms of retrieved target images have similar impressions to the impression of “muchal8.jpg”, because the query “Difference” emphasized the color features of “muchal8.jpg”. The highly-correlated impression words to the retrieved target images were “domestic”, “intimate”, “pastoral”, “nostalgic”, “lighthearted” and so on.

Query		Target Image	Histogram	Correlation		Target Image	Histogram	Correlation
 Difference (muchal8.jpg)	1	mucha18.jpg		0.647502	6	mucha22.jpg		0.464463
	2	mucha5.jpg		0.517954	7	mucha8.jpg		0.458932
	3	mucha19.jpg		0.49492	8	mucha31.jpg		0.455681
	4	mucha12.jpg		0.490948	9	mucha3.jpg		0.453159
	5	mucha26.jpg		0.473932	10	mucha2.jpg		0.420875


Highly-correlated Impression words & the Histograms	
	domestic, intimate, pastoral, nostalgic, lighthearted, restful, rich & luxurious, elegant

Figure 3: Retrieval results by the query of “Difference” of an image compared to the other images painted by the same author

These retrieval results show that the color features and impressions extracted by our image-query method using multiple images can not be extracted by a single image.

4.2 Experiment 2

First, we selected 8743 sample images of Ukiyo-e (traditional art work of Japan) from Tokyo Metropolitan Library [18] as an example of Eastern Art of the 19th – 20th Century for a query creation. Second, we set “Average of Meiji era (1868-1912)” and

“Difference of Bunsei period (1818-1830, late Edo era)” as query contexts. The number of collected sample images for the former was 3158 and the latter was 574.

Figure 4 shows the retrieval results by the query “Average” of an image set painted by different authors in the same period. In the top 10, the number of pictures exactly painted in Meiji era was 9, and “M341-017-14a.jpg” ranked in 3rd was the picture painted in the contiguous preceding period of Meiji era (Kaei period). The histogram of the query “Average of Meiji era” consists of strong and vivid colors. The highly-correlated impression words to the histogram of the query and the retrieved target images were “active”, “forceful”, “dynamic”, “energetic”, “bold” and so on. One of the reasons of these characteristic colorations in arts of Meiji era might be in the historical and cultural fact that new discovery of pigment ink of vivid red in this era. Another reason might be in the fact that the Meiji government had taken a policy of increasing wealth and military power, or that class systems and controls on culture were eliminated.



Query		Target Image	Correlation	Correctness		Target Image	Correlation	Correctness
 Average (Meiji era)	1	M347-046-08.jpg	0.93743	correct era	6	M348-011-05(03).jp	0.92370	correct era
	2	M347-002(02).jpg	0.93082	correct era	7	5721-C006-01.jpg	0.92346	correct era
	3	M341-017-14a.jpg	0.92848	preceding period	8	5721-C005(02).jpg	0.92324	correct era
	4	M348-029-02(03).jpg	0.92744	correct era	9	M348-029-03(03).jp	0.92250	correct era
	5	M648-007-03(01).jpg	0.92722	correct era	10	M247-015-02a.jpg	0.92214	correct era
Highly-correlated Impression words & the Histograms 								
active, forceful, dynamic, energetic, bold, intense, formal, sharp, mellow, earnest								

Figure 4: Retrieval results by the query of “Average” of an image set painted (by different authors) in the same period

Figure 5 shows the retrieval results by the query “Difference” of an image set painted by different authors in the same period, compared to the other image sets painted in the other periods. In the top 10, the number of pictures exactly painted in Bunsei period was 3, and pictures ranked in 1st, 2nd, 6th, 8th, 9th and 10th were the picture painted in the contiguous preceding and ensuing period of Bunsei period. These periods (Bunka:1804-1818, Tempo: 1830-1844) can be considered as the same cultural period because the time was under the same general’s sovereignty. The highly-correlated impression words to the histogram of the query and the retrieved target images were “chic”, “cozy & comfortable”, “cultured”, “delicate”, “gentle” and so on.



Query		Target Image	Correlation	Correctness		Target Image	Correlation	Correctness
 Difference (Bunsei period)	1	K1021-077.jpg	0.88502	preceding period	6	572-C023-01.jpg	0.86791	ensuing period
	2	N054-054(01).jpg	0.88436	preceding period	7	M338-009(01).jpg	0.86698	correct period
	3	M338-018-03.jpg	0.87102	correct period	8	M139-013-01(02).jpg	0.86058	ensuing period
	4	5721-C044.jpg	0.86996		9	5729-C012.jpg	0.85414	preceding period
	5	N029-004.jpg	0.86909	correct period	10	N054-054(03).jpg	0.85266	preceding period
Highly-correlated Impression words & the Histograms 								
chic, cozy & comfortable, cultured, delicate, gentle, maidenly, quiet, subtle, comfortable, tranquil								

Figure 5: Retrieval results by the query of “Difference” of an image set painted (by different authors) in the same period, compared to the other image sets painted in the other periods

5. Conclusion

In this paper, we have presented a dynamic image-query creation and metadata extraction method with semantic correlation computing between color-combinations and impressions of multiple image data. The created image-queries represent the impressions of colors of any specific author, time, culture, which are selected by a user as user's context. We also present an implementation method to create a semantic space which represents the relationships between colors and the impressions, and to highlight minor colors used in the image collection manufactured in a specific style, format or background. To fill a semantic gap between image features and user's context and use the varieties of historical and cultural semantics of color, we propose an image-query creation method for representing impressions with semantic correlation computing between color-combinations and impressions of multiple image data and apply the method to an image retrieval system especially for cultural properties. We have performed qualitative experiments with the image data, especially of cultural properties in digital archives to verify the feasibility of this image-query creation and metadata extraction method, and shown that our query creation method by multiple image data can extract appropriate impression which cannot be easily extracted from single image data.

As future work, we implement an on-the-fly image retrieval system with dynamic image-query creation and metadata extraction, and execute quantitative experiments on the large amount of image data.

References

- [1] H. Greenspan, S. Gordon and J. Goldberger, "Probabilistic models for generating, modeling and matching image categories", In Proc. of the International Conference on Pattern Recognition(ICPR'02), Quebec, August 2002.
- [2] Yixin Chen, James Ze Wang, Robert Krovetz, "Content-based image retrieval by clustering," Multimedia Information Retrieval, pp. 193-200, Nov. 2003.
- [3] M. L. Kherfi, D. Ziou, A. Bernardi: "Image Retrieval from the World Wide Web: Issues, Techniques, and Systems," ACM Computing Surveys (CSUR), Volume 36, Issue 1, March 2004.
- [4] Michael S. Lew, Nicu Sebe, Chabane Djeraba, LIFL, France, Ramesh Jain: "Content-based multimedia information retrieval: State of the art and challenges," ACM (TOMCCAP), Volume 2, Issue 1, pp. 1 - 19, February 2006.
- [5] Kitagawa, T. and Kiyoki, Y.: "The mathematical model of meaning and its application to multidatabase systems," Proceedings of 3rd IEEE International Workshop on Research Issues on Data Engineering Interoperability in Multidatabase Systems, April 1993, 130-135.
- [6] Yasushi Kiyoki, Takashi Kitagawa, Takanari Hayama: "A metadatabase system for semantic image search by a mathematical model of meaning," ACM SIGMOD Record, Volume 23 Issue 4, December 1994.
- [7] T. Kitagawa, T. Nakanishi, Y. Kiyoki: "An Implementation Method of Automatic Metadata Extraction Method for Image Data and Its Application to Semantic Associative Search," Information Processing Society of Japan Transactions on Databases, VOL.43, No. SIG12(TOD16), pp.38-51, 2002.
- [8] Kozaburo Hachimura: "Retrieval of Paintings Using Principal Color Information," Proc. CPR96, Vol.3, pp.130-134, 1996.
- [9] Wynne Hsu, Chua T.S., Pung H.K.: "An Integrated Color-Spatial Approach to Content-based Image Retrieval," ACM Multimedia 95 Electronic Proceedings, November 1995.
- [10] X. Chen and Y. Kiyoki, "A Visual and Semantic Image Retrieval Method," Information Modelling and Knowledge Bases, Vol. XVIII, pp.245-252, May 2007.
- [11] Faber Birren, Color and Human Response, John Wiley & Sons Inc, 1984
- [12] Faber Birren, COLOR, A SURVEY IN WORDS AND PICTURES, FROM ANCIENT MYSTICISM TO MODERN SCIENCE, New Hyde Park: University Books, 1963.
- [13] Johann Wolfgang von Goethe, Theory of Colours, trans. Charles Lock Eastlake, Cambridge, Massachusetts: The M.I.T. Press, 1982
- [14] Shigenobu Kobayashi, Color Image Scale, The Nippon Color & Design Research Institute ed., translated by Louella Matsunaga, Kodansha International, 1992.
- [15] Google Image Search, <http://images.google.com/>
- [16] Tokyo Metropolitan Library, <http://metro.tokyo.opac.jp/tml/tpic/>

Construction of Peer-to-Peer Systems for Knowledge Resource Distribution Using Overlay Clustering of Similar Peers

Huijun LI, Samir IBRADZIC, Xiao SHAO and Takehiro TOKUDA

Department of Computer Science, Tokyo Institute of Technology

Meguro, Tokyo 152-8552, Japan

{lhj, samir, shao, tokuda}@ft.cs.titech.ac.jp

Abstract. Peer-to-Peer (P2P) systems have been attractive solutions for resource distribution and sharing because of their self-organizing and fault-tolerant features. However, due to the decentralized nature, search efficiency is a challenging problem for large scale knowledge resource distribution P2P systems. In this paper, peer-similarity based P2P overlay clustering algorithm is proposed. Simulation results show that search efficiency using overlay clustering of similar peers is substantially improved.

Keywords. Peer-to-Peer systems, overlay clustering, similar peers

1. Introduction

Peer-to-Peer (P2P) systems are distributed systems without central control, in which nodes play the same role without distinguishing servers and clients. P2P systems have been proven to be effective to distribute and share resources. There are many features of P2P systems: self-organizing, fault-tolerant and low cost implementation.

Current knowledge resources are distributed over individuals, research groups and research domains, which make them difficult to share and utilize. Centralized solutions which build a large knowledge resource base to collect and share knowledge resources cost too much and might suffer from a single point of failure. In this paper, we propose a P2P based construction of knowledge resource distribution system (Fig. 1), which has the following advantages: (1) Knowledge resources are stored and managed by individuals. It is not necessary to invest in servers or storage equipments to store and manage a large number of resources. (2) Individuals can distribute their knowledge resources and access to other knowledge resources in the system.

Despite these advantages, we will address the problem of how to find the requested knowledge resource efficiently. We propose a clustering algorithm to improve search efficiency in unstructured P2P networks. In our proposal, we group peers with similar contents, and prioritize forwarding of the query messages inside clusters, in order to achieve a high hit ratio with low overhead.

The rest of this paper is organized as follows. In Section 2 we explain some related work and give an overview of existing P2P search methods. In Section 3, we illustrate the proposed clustering algorithm and evaluate the method by simulation in Section 4. Finally we conclude the paper in Section 5.

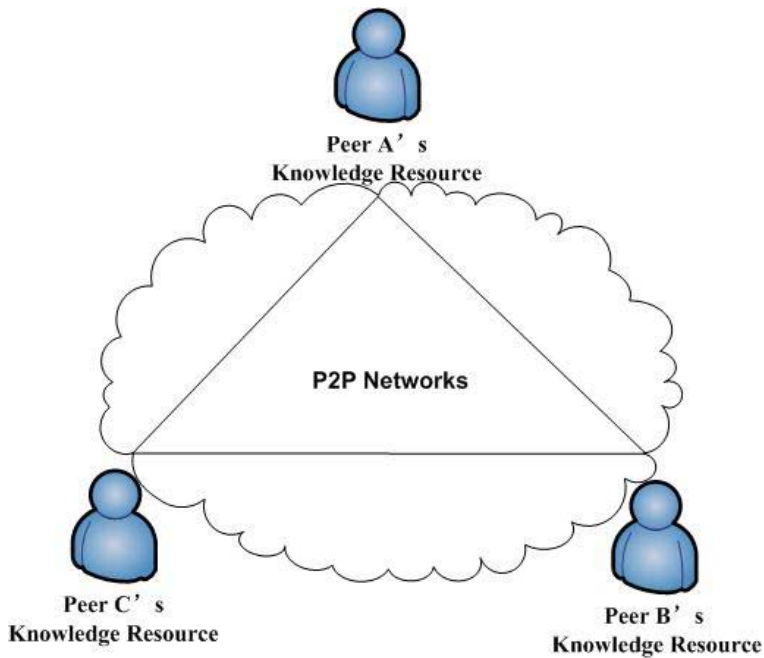


Figure 1. P2P networks for knowledge resource distribution.

2. Related Work

2.1. P2P Based Knowledge Resource Distribution Systems

Bibster [1] is a semantic-based bibliographic P2P system. In Bibster, peers advertise their expertise resources, which contain the set of topics on which the peers are expert. Other peers may accept the advertisements or not according to their local knowledge, thus create semantic links to their neighbors. These semantic links form a semantic overlay. It forms the basis for intelligent query routing. Edutella [2] propose RDF schema models to organize data. In Edutella, peers describe their capabilities and share these descriptions with other peers. So peers only route a query to those peers that are probably able to answer it. Bibster and Edutella focus on finding the right peers for querying, but they provide no means for semantic clustering of peers. pSearch [3] distributes document indices through the P2P networks based on document semantics generated by Latent Semantic Indexing [4]. pSearch is organized as a CAN (Content Addressable Network) [5]. In pSearch, all documents are mapped into the same semantic search space and the dimensionality of the overlay depends on the dimensionality of the vector. So if the vector has a high dimension, each peer has to know many neighbors.

2.2. Overview of P2P Search Methods

There are plenty of existing and proposed solutions for efficient search in P2P systems. Following the evolution of P2P systems, these search methods vary in implementation

according to type of P2P network overlays. Current proposals rely on partially modifying overlay networks in order to achieve better search performance or forcing these overlays to form a strict structure to achieve the same goal. Here we will mention and describe these P2P search methods mainly for the purpose of evaluation and comparison with our proposed solution.

2.2.1. Search in Early P2P Systems

First usable and pretty popular P2P system was Napster [6]. This system was a highly centralized one, with all of the nodes sending all queries to a central server, which acted as a database with indices of all content of all of the peers in the network. This kind of structure was known as centralized P2P system, which, by maintaining centralized database, could provide search guarantees and enable complex lookups, but could not be feasible enough and did not provide enough scalability for demanding and rapidly growing P2P users. Also, centralization raised the maintenance costs and posed the risk as single point of failure in both technical and legal aspects. Therefore, these systems were outdated and replaced by others that actually do scale with growing demands.

2.2.2. Flooding Search

Soon after the shortcomings of heavily centralized P2P systems became obvious, decentralized P2P systems appeared. Goal was to completely eliminate any single point of failure, therefore, any central control entity was not allowed, all peers in the P2P overlay was equal, and overlay network was random. Typical representative of this approach was Gnutella [7] network (Gnutella 0.4, early implementation). Gnutella used flooding search, in which each node, when performing a lookup query, would flood all of its neighboring nodes with query, and then these nodes would pass this query to their own neighbors, and so on, until all the corners of the P2P network overlay have been checked for query (flooded) or the number of passes reach certain TTL threshold value. This was quite effective for relatively small scale P2P networks, but did not guarantee that all existing information that matches a query would be found. If these networks grow into the size of a millions of peers (as happened with Gnutella at the peak of its popularity), this approach proved to be almost disastrous. If we presume some few thousands of the nodes starting sending queries at certain point of time, all of these queries will eventually multiply exponentially in order to progress throughout the whole network. This situation creates excessive amounts of network traffic, and uses significant processing power to route queries, just to search for something that might not be available at all. It was clear that flooding method scales well only up to some point, and needs replacement or improvement. Actually, many improvements were proposed, such as Gnutella 0.6 protocol [8], and our method is one of these that restricts and optimizes flooding in certain way, mostly by optimizing overlay structure.

2.2.3. DHT (Distributed Hash Tables) Lookup Search

A hash table seems as a natural solution for implementing a lookup or search protocol. By providing some identifier (hash key), the corresponding data can be retrieved. Data stored in hash tables must be identified using unique numeric key. A hashing function converts the data identifier into this key. In the case of P2P system, the hash table is distributed among the peers, with each peer storing a part of it. Usually, consistent hashing was used to ensure uniform distribution of load among the peers. DHT method

is based on structured P2P overlay, both data and the topology of the overlay are highly structured and controlled, ensuring access guarantees. Unfortunately, as the mapping of data elements to hash keys is based on unique data identifiers, only these identifiers can be used to retrieve the data. Keyword searches and more complex queries are not possible to implement easily on typical DHT systems, which include: Chord [9], CAN [5], Pastry [10] and Tapestry [11].

2.2.4. Semantic Overlay Search

In order to improve the search efficiency and scalability in unstructured P2P networks, the concept of Semantic Overlay Networks (SONs) [12] has been proposed. General idea of this approach is to reorganize or establish new P2P network overlay according to semantic information inside peers in order to optimize search. Nodes with semantically similar content are clustered. Queries are forwarded to the SONs which are most probable to answer them, thus increase the hit ratio and reduce the number of messages to answer queries. But how to create SONs in a P2P manner is a problem.

Recently there are several approaches proposed to use SONs to improve search in P2P systems. For example, [13] proposes a method based on rearranging the connections between peers to link friend peers to each other. In [14], associative overlays are proposed, which are formed by peers that have provided answers to previous queries. Several other approaches to create SON over structured P2P systems have also been proposed. In this paper, we concentrate on unstructured P2P systems.

3. P2P Overlay Clustering Based on Peer Similarity

3.1. Overview

Usually researchers perform search for bibliographic information through university libraries or via web search engines such as Google. This example of knowledge resource distribution system is used for computer science researchers to share the bibliographic information between each other in a P2P environment. The bibliographic information distributed in the system is classified according to ACM CCS (computing classification system) [15]. Every piece of bibliographic information belongs to one or more categories in the ACM CCS. Among the large number of distributed resources, a way to find the requested resource efficiently is important. For example, when a researcher is asking for the papers on the topic of ACM Topics/information systems/database management, the researcher whose knowledge resource is about ACM Topics/information systems/database management is more likely to answer the query than a researcher whose knowledge resource is about ACM Topics/hardware/control design systems. So if we can send queries to only those peers which already have existing resources on ACM Topics/information systems/database management, the hit ratio will be improved and number of query messages will be reduced.

The basic idea of our work is to organize peers into clusters, in which peers have similar resources. Once a query has reached the right cluster of peers, it will be more likely to be answered inside the cluster. Clustering similar peers is based on the community structure of networks [16]. Community structure is a kind of property that groups of nodes have a high density of connections within them, while connections between groups are sparse.

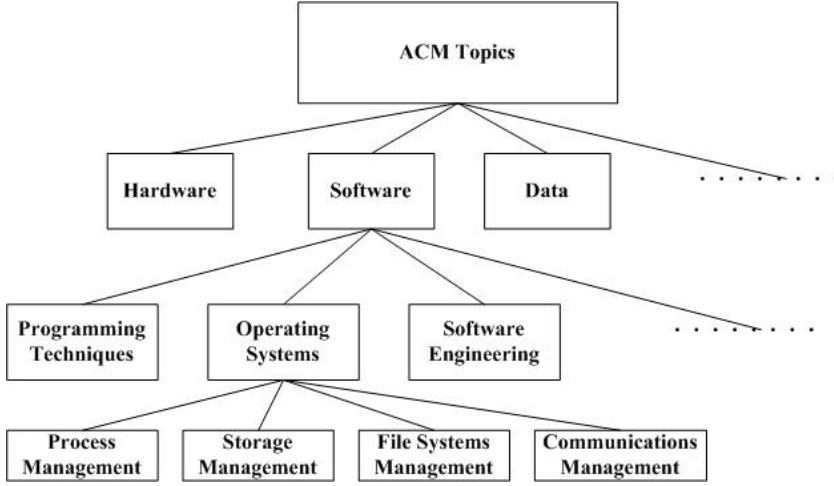


Figure 2. Part of the complete ACM CCS.

3.2. Peer Similarity Computing

Before introducing the clustering of the similar peers, we will first illustrate how to measure the similarity between peers which is the basis for peer clustering. The knowledge resources distributed in P2P networks are classified according to ACM CCS. ACM CCS is a hierarchical classification, a tree-like structure by its nature (Fig. 2).

According to [17], similarity of topics in the classification tree can be calculated: The similarity between two topics is between $[0, 1]$. When we assign values to parameters $\alpha = 0.2$, $\beta = 0.6$, formula Eq. (1) can yield best result:

$$sim_{topic}(t_1, t_2) = \begin{cases} e^{-\alpha l} * \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} (t_1 \neq t_2) \\ 1 (t_1 = t_2) \end{cases} \quad (1)$$

where t_1 and t_2 are topics in the classification tree, l is the minimum length of path connecting two topics, h is the depth of subsumer (the first topic in the hierarchical tree which contains the topic t_1 and t_2) in the hierarchy tree.

In order to calculate how similar the peers are, we need a function to compute the similarity between peers. Assuming there are two peers P and Q, each peer have resources on some topics. The similarity of peer P and Q can be calculated using Eq. (2), Eq. (3) and Eq. (4):

$$Sim(P, Q) = \sum_{j=1}^{|Q|} \sum_{i=1}^{|P|} [sim(t_i, t_j) \times (W_i \times W_j)] \quad (2)$$

$$W_i = N_i / \sum_{k=1}^{|P|} N_k \quad (3)$$

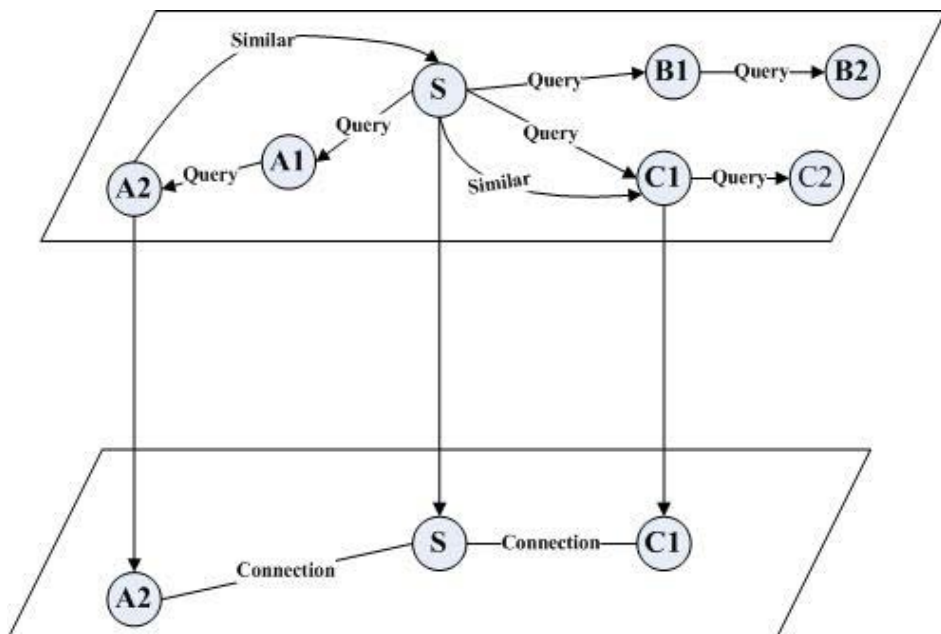


Figure 3. P2P semantic overlay model.

$$W_j = N_j / \sum_{k=1}^{|Q|} N_k \quad (4)$$

where $|P|$ and $|Q|$ refer to the number of topics which peer P and Q have, t_i and t_j represent the topics, N_i and N_j are the number of resources which are classified into topic t_i and t_j , W_i and W_j are the weight of topic t_i , t_j in peer P and Q.

3.3. P2P Semantic Overlay and Clustering

3.3.1. Semantic Overlay Building

In order to form clusters in which peers with similar content are grouped together, we build a semantic overlay network above the original P2P overlay network.

The semantic overlay network is described as a graph $G = (E, V)$ where V is the set of nodes, E is the set of edges and there is an edge $\{(i, j) \in E \mid i, j \in V\}$ where similarity between peer P_i and peer P_j is larger than the threshold.

The process of building semantic overlay is described as follows: When a peer P joins the P2P network, it has a set of neighbors. Peer P sends messages with a given TTL value to all its neighbors. Neighbors reply to the query and send their own resource profiles to P and $TTL - 1$. If the similarity of P and its neighbors, for example P_i is larger than the threshold, there is a link between P and P_i , and P P_i is the semantic neighbor of P. If $TTL > 0$, P_i forwards query messages to its neighbors and collects the resource profiles of them. The process continues until $TTL = 0$. Thus peers can find their semantic neighbors and the semantic overlay can be built (Fig. 3).

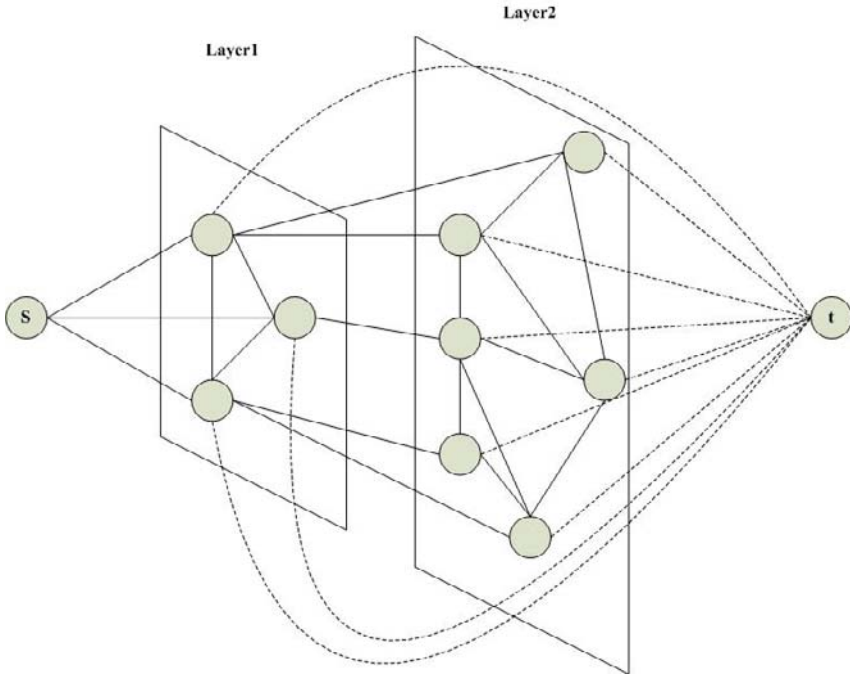


Figure 4. Collecting graph information and augmenting the graph.

3.3.2. Semantic Overlay Clustering

In Gnutella-like unstructured P2P networks, nodes are connected randomly, and because of this kind of topology, we can not distinguish the community structure by topology or some other property directly. In [18], the property of community structure was identified, in semantic overlay that peers with semantically similar contents are “clustered” together, and peers are self-organized into groups within which connections are dense, but between which connections are sparse.

Discovering community structure in networks is a kind of NP-complete (“non-deterministic polynomial time”) problem. But in a P2P network the whole graph of the overlay network is unknown. So the general methods for discovering community structure are not applicable. In [19], a method to construct scalable P2P networks by self-clustering is proposed. The max-flow min-cut algorithm [20] is applied to calculate clusters. Sharing the same idea, we cluster P2P semantic overlay built above the existing P2P overlay.

In the semantic overlay, the weight of edge (i, j) equals to the times of successful download between i and j . We set the link degree of a node as a threshold parameter. When the link degree of a peer reaches the threshold, the peer will begin to run the clustering algorithm as the source S to create clusters. The source node S first gets its neighbors (Layer1 nodes in Fig. 4) according to its local semantic neighbor information. Then, nodes in Layer1 start to gather the semantic information about their neighbors (Layer2 nodes in Fig. 4). If there are enough nodes acquired, the process stops. Finally, we add a virtual sink node t , and all the nodes connect to t with a constant capacity value link. We run the max-flow min-cut algorithm on the augmented graph, and the cluster is formed out of the set of the nodes connected the source S .

Table 1. Parameters in simulation

Parameter	Meaning
Semantic overlay TTL	TTL for generate semantic overlay
Clustering threshold	The value of semantic link degree to run the clustering algorithm
Virtual link weight	The capacity of the links to the virtual sink t
Similarity threshold	Threshold of peer similarity

3.4. Query Routing Algorithm

When a peer receives a query, it checks the similarity of the query with its own resource profile. If they are similar, it searches in its own repository and forwards the query to other members in the same cluster. Otherwise, before forwarding the query, it checks the similarity between the query and its neighbors' resource profiles. The query will be sent to the neighbors which are similar to the query and be forwarded in the cluster of these neighbors. The similarity between query and peer can be calculated using Eq. (5):

$$Sim(query,P)=\max\{Sim(t_q,t_i)\times W_i\} \tag{5}$$

where t_i and t_q represent the topics of peer P and query, W_i is the weight of topic t_i in peer P.

4. Simulation Results

We performed simulations to evaluate the performance of proposed method. We first implemented a Gnutella style overlay network and apply our algorithm to it. The overlay network topology is a random graph and the resource distribution is uniform distribution. We list some parameters in Table 1.

To evaluate the proposed method, we use the number of messages and hit rate for comparison. Figure 5 shows that search on our cluster based overlay produces fewer messages than on Gnutella style overlay does. In Fig. 6 we compare the hit rate of two methods when the network size is 5000 and it can be seen from the graph that hit rate of proposed method is higher than Gnutella style overlay does. Due to the size of overlay network, the two lines of hit rate close to each other after the query messages are forwarded 4 hops.

5. Conclusion

We have proposed a method to construct P2P systems for knowledge resource distribution. Users can join the system to use knowledge resources as well as sharing their own resources. By evaluating the similarity of peers, we cluster similar peers and improve the search performance of P2P networks.

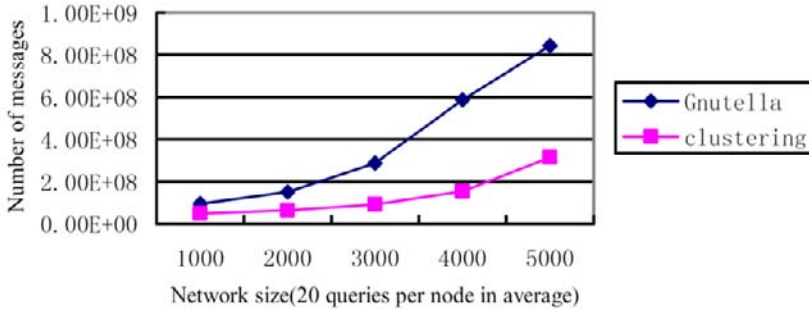


Figure 5. Comparison of messages in different network scale.

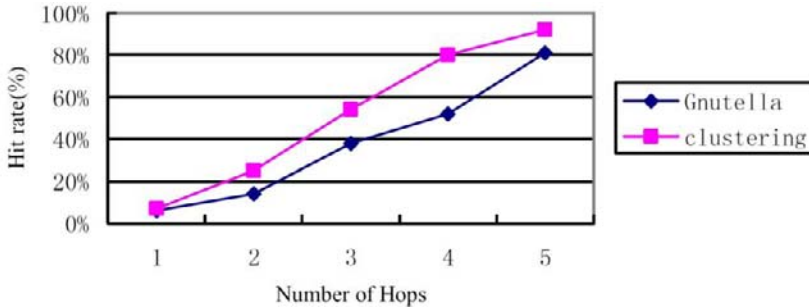


Figure 6. Comparison of hit rate.

References

- [1] Haase, P., Broekstra, J., Ehrig, M., Menken, M., Mika, P., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S. and Tempich, C.: Bibster: a semantic-based bibliographic peer-to-peer system. In Proceedings of ISWC, 2004.
- [2] Nejdl, W., Wolpers, M., Siberski, W. and Schmitz, C.: Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks. In Proceeding of WWW, 2003.
- [3] Tang, C., Xu, Z. and Dwarkadas, S.: Peer-to-peer information retrieval using self-organizing semantic overlay networks. In Proceeding of ACM SIGCOMM, 2003.
- [4] Berry, M. W., Drmac, Z. and Jessup, E. R.: Matrices, vector spaces, and information retrieval. SIAM Review, Vol. 41, No. 2, pp. 335-362, 1999.
- [5] Ratnasamy, S., Francis, P., Handley, M., and Karp, R.: A scalable content-addressable network. In Proceedings of ACM SIGCOMM, 2001.
- [6] Napster webpage: <http://www.napster.com/>.
- [7] Gnutella webpage: <http://www.gnutella.com/>.
- [8] Gnutella Protocol Specification webpage: <http://gnutella-specs.rakjar.de/>.
- [9] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of ACM SIGCOMM, 2001.
- [10] Rowstron, A. and Druschel, P.: Pastry: Scalable, distributed object location and routing for large scale peer-to-peer systems. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), 2001.
- [11] Zhao, B. Y., Kubiawicz, J., and Joseph, A. D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, Computer Science Division, University of California at Berkeley, 2001.
- [12] Crespo, A. and Garcia-Molina, H.: Semantic overlay networks for P2P systems. Technical Report, Stanford University, 2002.
- [13] Cohen, E., Fiat, A. and Kaplan, H.: Associative search in peer to peer Networks: harnessing latent semantics. In Proceedings of INFOCOM'03, 2003.

- [14] Parreira, J. X., Michel, S. and Weikum, G.: p2pDating: Real life inspired semantic overlay networks for web search. In Proceedings of SIGIR Workshop on heterogeneous and distributed information retrieval, 2005.
- [15] ACM 1998 Classification website: <http://www.acm.org/class/1998>.
- [16] Newman, M. E. J.: The structure and function of complex networks. *SIAM Review*, Vol. 45, No. 2, pp. 167-256.
- [17] Li, Y., Bandar, Z. A. and McLean, David.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 4, pp. 871-881, 2003.
- [18] Chen, H., and Jin, H.: Identifying community structure in semantic peer-to-peer networks. Second International Conference on Semantics, Knowledge, and Grid, 2006.
- [19] Shao, X. and Tokuda, T.: A construction of scalable P2P networks by self-clustering. 13th International Conference on Telecommunications, pp. 17-20, 2006.
- [20] Ford, L. R. and Fulkerson, D. R.: Maximal flow through a network. *Canadian Journal of Mathematics*, Vol. 8, pp. 399-404, 1956.

Co-Design of Web Information Systems Supported by SPICE

Gunar FIEDLER^b, Hannu JAAKKOLA^a, Timo MÄKINEN^a,
Bernhard THALHEIM^b and Timo VARKOI^a

^a *Tampere University of Technology, P.O. Box 300, FI-28101 Pori, Finland*
{hannu.jaakkola, timo.makinen, timo.varkoi}@tut.fi

^b *Department of Computer Science, Kiel University, Olshausenstr. 40, 24098 Kiel,
Germany*
{fiedler,thalheim}@is.informatik.uni-kiel.de

Abstract. Web information systems (WIS) augment classical information systems by modern Web technologies. They require at the same time a careful development and support for the interaction or story spaces beside the classical support for the working space of users. These dimensions complicate the system development process. This paper shows how classical advanced methodologies can be carefully enhanced. We evaluated the Co-Design approach to information systems development according to the ISO/IEC 15504 Framework for Process Assessment (SPICE) and derived the potential and deficiencies of this approach. This evaluation has been used for a managed Co-Design methodology. Since WIS constantly change and evolve the development process never ends. We develop as a solution an optimization procedure and a runtime environment for the Co-Design approach that allows to cope with such changes, with evolution and extension of WIS and that demonstrates the new system facilities whenever a change is mastered.

1. Introduction

Planning, developing, distributing, and maintaining sophisticated large-scaled systems is one of the core competencies in software engineering. Properly working systems provide valuable assets to users as well as operators while erroneous, incomplete, and misused software causes losses in economical, technical, or social ways as systems become more and more ubiquitous.

Information systems are considered to be complex systems with a deep impact on the people's daily life. *Web information systems* (WIS) are systems providing information to users by utilizing Web technologies. Usually, WIS are data-intensive applications which are backed by a database. While the development of information systems is seen as a complex process, *Web information systems engineering* (WIS-E) adds additional obstacles to this process because of technical and organizational specifics:

- WIS are open systems from any point of view. For example, the user dimension is a challenge. Although purpose and usage of the system can be formulated in advance, user characteristics cannot be completely predefined. Applications have to be intuitively usable because there cannot be training courses for the users. Non-functional properties of the application like ‘nice looking’ user interfaces are far more important compared with standard business software. WIS-E is not only restricted to enterprises but is also driven by an enthusiastic community fulfilling different goals with different tools.
- WIS are based on Web technologies and standards. Important aspects are only covered by RFCs because of the conception of the Internet. These (quasi-)standards usually reflect the ‘common sense’ only, while important aspects are handled individually.
- Looking at the complete infrastructure, a WIS contains software components with uncontrollable properties like faulty, incomplete, or individualistically implemented Web browsers.
- Base technologies and protocols for the Web were defined more than 10 years ago to fulfill the tasks of the World Wide Web as they had been considered at this time. For example, the HTTP protocol was defined to transfer hypertext documents to enable users to browse the Web. The nature of the Web changed significantly since these days, but there were only minor changes to protocols to keep Compatibility alive which is considered to be “indispensable”. Today, HTTP is used as a general purpose transfer protocol which is used as the backbone for complex interactive applications. Shortcomings like statelessness, loose coupling of client and server, or the restrictions of the request-response communication paradigm are covered by proprietary and heavy-weight frameworks on top of HTTP. Therefore, they are not covered by the standard and handled individually by the framework and the browser, e.g., session management. Small errors may cause unwanted or uncontrollable behavior of the whole application or even security risks.

WIS can be considered from two perspectives: the system perspective and the user perspective. These perspectives are tightly related to each other. We consider the presentation system as an integral part of WIS. It satisfies all user requirements. It is based on real life cases. Software engineering has divided properties into functional and non-functional properties, restrictions and pseudo-properties. This separation can be understood as a separation into essential properties and non-essential ones. If we consider the dichotomy of a WIS then this separation leads to a far more natural separation into information system requirements and presentation systems requirements. The system perspective considers properties such as performance, efficiency, maintainability, portability, and other classical functional and non-functional requirements. Typical presentation system requirements are usability, reliability, and requirements oriented to high quality in use, e.g., effectiveness, productivity, safety, privacy, and satisfaction. Safety and security are also considered to be restrictions since they specify undesired behavior of systems. Pseudo-properties are concerned with technological decisions such as language, middleware, operating system or are imposed by the user environment, the channel to be used, or the variety of client systems.

WIS must provide a sophisticated support for a large variety of users, a large variety of usage stories, and for different (technical) environments. Due to this flexibility the development of WIS differs from the development of information systems by careful elaboration of the application domain, by adaptation to users, stories, environments, etc.

Classical software engineering typically climbs down the system ladder to the implementation layer in order to create a productive system. The usual way in today's WIS development is a manual approach: human modelling experts interpret the specification to enrich and transform it along the system ladder. This way of developing specifications is error-prone: even if the specification on a certain layer is given in a formal language, the modelling expert as a human being will not interpret it in a formal way. Misinterpretations, misunderstandings, and therefore the loss of already specified system properties is the usual business.

The paper is organized as follows: In Section 2 we discuss existing approaches and methodologies for WIS development, especially the Co-Design approach with its integrated view on structure, functionality, interactivity, and distribution. The presented methodologies miss central specifics in today's WIS development. No methodology is of value if it is not supported and enforced by an organizational and technical infrastructure. The section ends with a short discussion of the international standard ISO/IEC 15504, which provides a framework for the assessment of software processes. Section 3 partially presents the results of an assessment of the WIS Co-Design approach to enable the implementation of ideas from ISO/IEC 15504 to develop the WIS Co-Design approach towards managed WIS engineering. Section 4 shows the application of principles of SPICE to WIS development within the Co-Design approach to prepare WIS-E processes to move towards the 'Optimizing' level.

2. Background and Related Work

Several methodologies and architectures were developed to cope with information systems engineering in general and WIS-E in particular. [KPRR03] provides an overview over concepts, methods, and tools in WIS-E as well as the relationships between classical software engineering and web development.

2.1. Classical (Web) Information Systems Methodologies

ARIS (*Architecture of Integrated Information Systems*, [Sch92]) defines a framework with five views (functional, organizational, data, product, controlling) and three layers (conceptual ('Fachkonzept'), technical ('DV-Konzept'), and implementation). ARIS was designed as a general architecture for information systems in enterprise environments. Therefore, it is too general to cover directly the specifics of Web information systems as they were mentioned in Section 1 and needs to be tailored.

The *Rational Unified Process* (RUP, [Kru98]) is an iterative methodology incorporating different interleaving development phases. RUP is backed by sets of development tools. RUP is strongly bound to the Unified Modelling Language (UML). Therefore, RUP limits the capabilities of customization. Like ARIS, RUP does not address the specifics of WIS-E. A similar discussion can be made for other general purpose approaches from software engineering [HSSM⁺04].

OOHDM [SR98] is a methodology which deals with WIS-E specifics. It defines an iterative process with five subsequent activities: requirements gathering, conceptual design, navigational design, abstract interface design, and implementation. OOHDM considers Web Applications to be hypermedia applications. Therefore, it assumes an inherent navigational structure which is derived from the conceptual model of the application domain. This is a valid assumption for data-driven (hypermedia-driven) Web applications but does not fit the requirements for Web information systems with dominating interactive components (e.g., entertainment sites) or process-driven applications. There are several other methodologies similar to OOHDM. Like OOHDM, most of these methodologies agree in an iterative process with a strict top-down ordering of steps in each phase. Surprisingly, most of these methodologies consider the implementation step as an 'obvious' one which is done by the way, although specifics of Web applications cause several pitfalls for the inexperienced programmer especially in the implementation step. Knowledge management during the development cycles is usually neglected.

There are several methodologies that cope with personalization of WIS. For example, the HERA methodology [HBFV03] provides a model-driven specification framework for personalized WIS supporting automated generation of presentation for different channels, integration and transformation of distributed data and integration of Semantic Web technologies. Although some methodologies provide a solid ground for WIS-E, there is still a need for enhancing the possibilities for specifying the interaction space of the Web information system, especially interaction stories based on the portfolio of personal tasks and goals.

2.2. Co-Design of Web Information Systems

We distinguish a number of facets or views on the application domain. Typical facets to be considered are business procedure and rule facets, intrinsic facets, support technology facets, management and organization facets, script facets, and human behavior. These facets are combined into the following aspects that describe different separate concerns:

- The *structural* aspect deals with the data which is processed by the system. Schemata are developed which express the characteristics of data such as types, classes, or static integrity constraints.
- The *functional* aspect considers functions and processes of the application.
- The *interactivity* aspect describes the handling of the system by the user on the basis of foreseen stories for a number of envisioned actors and is based on media objects which are used to deliver the content of the database to users or to receive new content.
- The *distribution* aspect deals with the integration of different parts of the system which are (physically or logically) distributed by the explicit specification of services and exchange frames.

Each aspect provides different modelling languages which focus on specific needs. While higher layers are usually based on specifications in natural language, lower layers facilitate formally given modelling languages. For example, the classical WIS Co-Design approach uses the Higher-Order Entity Relationship Modelling language [Tha00] for

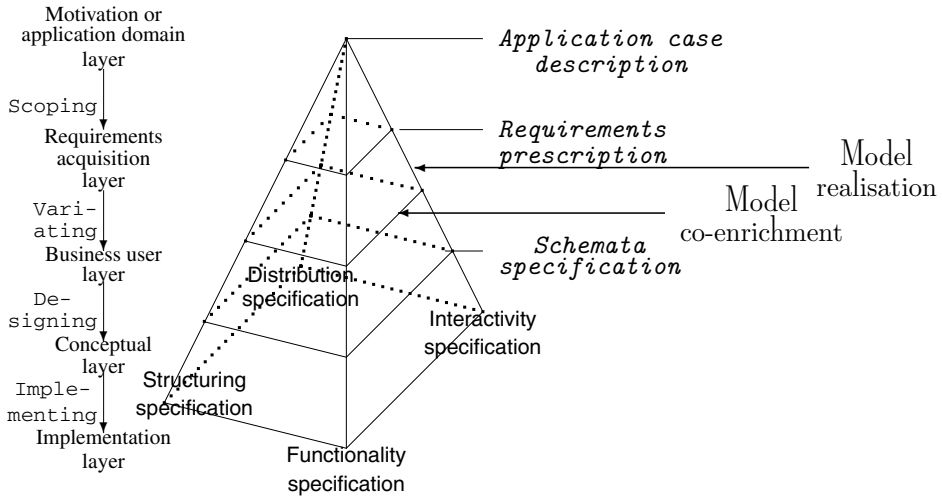


Figure 1. Abstraction Layers and Model Categories in WIS Co-Design.

modelling structures, transition systems and Abstract State Machines [BS03] for modelling functionality, Sitelang [TD01] for the specification of interactivity, and collaboration frames [Tha03] for expressing distribution. Other languages such as UML may be used depending on the skills of modelers and programmers involved in the development process.

A specification of a WIS consists of a specification for each aspect such that the combination of these specifications (the integrated specification) fulfills the given requirements. Integrated specifications are considered on different levels of abstraction (see Fig. 1) while associations between specifications on different levels of abstraction reflect the progress of the development process as well as versions and variations of specifications.

Unfortunately, the given aspects are not orthogonal to each other in a mathematical sense. Different combinations of specifications for structure, functionality, interactivity, and distribution can be used to fulfill given requirements while the definition of the ‘best combination’ relies on non-functional parameters which are only partially given in a formal way. Especially the user perspective of a WIS contributes many informal and vague parameters possibly depending on intuition. For example, ordering an article in an online shop may be modelled as a workflow. Alternatively, the same situation may be modelled by storyboards for the dialog flow emphasizing the interactivity part. This principle of designing complex systems is called *Co-Design*, known from the design process of embedded systems where certain aspects can be realized alternatively in hardware or software (Hardware Software Co-Design). The Co-Design approach for WIS-E developed in [Tha00,Tha03,Tha04] defines the modelling spaces according to this perception.

We can identify two extremes of WIS development. *Turnkey development* is typically started from scratch in a response to a specific development call. *Commercial off-the-shelf development* is based on software and infrastructure whose functionality is decided

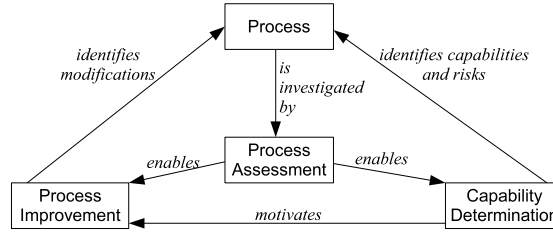


Figure 2. Process Assessment, Improvement, and Capabilities.

upon by the makers of the software and the infrastructure than by the customers. A number of software engineering models has been proposed in the past: waterfall model, iterative models, rapid prototyping models, etc. The Co-Design approach can be integrated with all these methods.

At the same time, developers need certain flexibility during WIS engineering. Some information may not be available. We need to consider feedback loops for redoing work that has been considered to be complete. All dependencies and assumptions must be explicit in this case.

2.3. Aligning Co-Design and the ISO/IEC 15504 Framework for Process Assessment

There are several approaches for improving a software process, e.g. modelling, assessment, measurement, and technology adoption [SO98]. The approaches supplement each other, but one of them is usually in a dominating position. Process assessment is a norm-based approach, which usually leads to evolutionary process improvement [AAMN01]. The starting point for software improvement actions is the gap between the current state of an organization and the desired future state [GM97]. These two states can be characterized using a norm for good software practices. The most popular among such norms are the Software Engineering Institutes CMMI (Capability Maturity Model Integration) [CMM06] and SPICE (Software Process Improvement and Capability dEtermination) [ISO05], developed by international standardization bodies (ISO and IEC). The current state of software practice can be evaluated with the help of process assessments, which is a disciplined appraisal of an organizations software process by utilizing a norm: the software process assessment model. SPICE (the technical report ISO/IEC TR 15504) has provided a framework for the assessment of software processes since 1998. The term SPICE originally stands for the initiative to support the development of a standard for software process assessment. The international standard ISO/IEC 15504, with five parts, has been published in 2003–2006 to replace the set of technical reports. Besides including a norm for a software process, SPICE can also be regarded as a meta-norm that states requirements for software process assessment models. Figure 2 (from [Bal98, p. 378]) depicts the context, where SPICE is applied.

Process capability is a characterization of the ability of a process to meet current or projected business goals [ISO04]. Part 2 of the standard defines a measurement framework for the assessment of process capability. In the framework, process capability is defined on a six point ordinal scale (0-5) that enables capability to be assessed from Incomplete (0) to Optimizing (5). The scale represents increasing capability of the implemented process. The measure of capability is based upon a set of process attributes, which define particular aspects of process capability [ISO03]. For example, Level 1 (“Performed Process”) requires that an implemented process achieves its process specific purpose. The level is characterized by one process attribute (Process Performance), which presumes certain outcomes from the process. Level 2 process is a Performed Process, which is also Managed i.e., implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained. Level 2 requirements are described by two process attributes: Performance Management and Work Product Management [ISO03]. The Co-Design methodology was examined using SPICE criteria with the aim to generate improvement ideas for the methodology. A brief description of the initiative can be found in [FTJ⁺07]. The following section illustrates the implementation of basic Process Performance requirements in Co-Design for WIS to form a basis for managed WIS engineering.

2.4. Requirements to Managed and Optimizable Co-Design for WIS

SPICE requires for managed development processes that the implemented process achieves its process purpose (SPICE level 1). SPICE level 2 is achieved if the process is well-specified and is implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained. Therefore, managed engineering is based on performance management and on work product management. Performance management requires that

- objectives for the performance of the process are identified,
- performance of the process is planned and monitored,
- performance of the process is adjusted to meet plans,
- responsibilities and authorities for performing the process are defined, assigned and communicated,
- resources and information necessary for performing the process are identified, made available, allocated and used, and
- interfaces between the involved parties are managed to ensure both effective communication and also clear assignment of responsibility.

Work product management has to be well-defined and well-implemented. Requirements for the work products of the process must be defined as well as requirements for documentation and control of the work. Work products must be appropriately identified, documented, and controlled. Work products are going to be reviewed in accordance with planned arrangements and adjusted as necessary to meet requirements.

3. Orchestration of WIS Development for Managed Engineering

Developing WIS using the Co-Design approach can be seen as an orchestration of different specifications. Orchestration uses the metaphor to music. It denotes the arrangement of a musical composition for performance by an orchestra. It may also denote harmonious organization, i.e. through orchestration of cultural diversities.

We show now in the sequel how orchestration of WIS development leads to managed WIS engineering. Due to space limitations we restrict on the work products and activities of the first process: Application domain description and requirements statement. It aims in describing the application perspective, e.g., the subject world, the usage world, and the intentions of the WIS according to [Poh96,RSE04,Rol06]. It results in a general statement of requirements. Requirements engineering aims in elicitation of requirements within the system environment, exploration of system choices, complete extraction of functional and non-functional requirements, conflict detection and resolution, documentation and negotiation of agreed requirements, and in providing a framework for WIS evolution.

(WIS-E 1): Application Domain Description and Requirements Statement

The most important outcome of application domain engineering is an application domain model and its associated application domain theory. The main activity is to gather from application domain business users, from literature and from our observations the knowledge about the application domain. It is combined with validation, i.e. the assurance that the application domain description commensurates with how the business users view the application domain. It also includes the application domain analysis, e.g., the study of application domain (rough) statements, the discovery of potential inconsistencies, conflicts and incompleteness with the aim of forming concepts from these statements.

Process Purpose: Goals and Subject

Goals and subject	Application domain description Agreement for development Project scope: Milestones, financial issues Clarification of development goals (intentions, rationale) Sketch of requirements
-------------------	--

Process Outcomes: Work Products as Process Results

The work product is a result or deliverable of the execution of a process and includes services, systems (software and hardware) and processed materials. It has elements that satisfy one or more aspects of a process purpose and may be represented on various media (tangible and intangible).

Documents of the application domain layer are HERM [Tha00] concept maps, HERM functionality feature descriptions, the DistrLang distribution specification resulting in contract sketches which include quality criteria, and the Sitelang interactivity specification of the application story with main application steps. The documents are combined within the *Stakeholder contract* ('Lastenheft') and the feasibility study. Additionally, a number of internal documents are developed such as life case studies, description of intentions, context specification, and user profiles and portfolios.

Developed documents Official and contracting section	Stakeholder contract: goal information, concept sketch, product functionality, story space, views on product data, view collaboration sketch Comparison with products of competitors Evaluation of development costs
Developed documents Application domain description section	Information analysis missions and goals of the WIS, brand of the WIS general characterization of tasks and users general characterization of content and functions description of WIS context Intensions of the web information system, catalog of requirements scenarios, scenes, actions, context and life cases user profiles and user portfolio actor profiles and actor portfolio, personas Business rule model and storyboards scenarios, scenes, and actions life cases and context WIS utilization portfolio scenarios, activities supporting content and functions non-functional requirements, context space Metaphor description base metaphors, overlay metaphors metaphor integration and deployment
Developed documents Internal section	Planning of goals, development strategy and plan, quality management Development documents on product components and quality requirements with base practices, generic practices and capabilities, estimation of efforts

Base Activities and Steps

We envision three base activities: (1) Development of application domain description, (2) Development of the stakeholder contract, and (3) Development of the internal documents such as description of product components and quality requirements with base activities, generic practices and capabilities and estimation of efforts, etc.

We demonstrate the base activities for the first one. We use the Sitelang specification language [TD01] and the conceptual framework of [Tha05].

Development of application domain description	<ol style="list-style-type: none"> 1. Analyze strategic information <ul style="list-style-type: none"> Specify mission and brand Characterize in general tasks and users Characterize in general content and functions Describe WIS context 2. Derive intensions of the WIS, <ul style="list-style-type: none"> Obtain general requirements Extract life cases Describe scenarios, scenes, actions, and context Describe user profiles and user portfolio Derive actor profiles and actor portfolio, personas 3. Extract business rule model and storyboards <ul style="list-style-type: none"> Develop scenarios, scenes, and actions Specify life cases and context Eliciting metaphors
	<ol style="list-style-type: none"> 4. Revise business rules of the application <ul style="list-style-type: none"> possibly with reorganization models 5. Compare new business rules with known visions 6. Compare with products of competitors 7. Derive WIS utilization <ul style="list-style-type: none"> Describe scenarios to be supported Describe activities based on word fields Describe supporting content Describe supporting functions Describe non-functional requirements Describe the context space 8. Develop metaphor <ul style="list-style-type: none"> Describe base and overlay metaphors Find metaphor integration Develop templates for deployment
Precondition	Contracted collaboration of all partners Real necessity
Postcondition	Descr. of application domain accepted and consistent New business rule model accepted

Information analysis is based on WIS storyboard pragmatics. We usually start with templates that specify the brand of the WIS by the four dimensions provider, content, receiver, and main actions. The brand is based on the mission containing general statements on the content, the user, the tasks, purposes, and benefits. The content indication may be extended on the basis of the content item specification frame and by a rough description of context based on the general context template. The brand description is compared with the description of tasks to be supported. Tasks are specified through the general characterization template. The specification of the intention of the WIS is based on the designation of goals of the WIS. Additionally we integrate metaphors. They must be captured as early as possible. Metaphors may be based on the style guides that are provided by the customer.



Figure 3. Story Space for the Photo Gallery.

4. SPICEing Co-Design: Towards Optimizing Processes

Every methodology is only valuable if it is supported and enforced by the technical and organizational environment. The methodology itself is only able to provide a frame for the quality management of each development process and the knowledge transfer between development processes.

Imagine, you are requested to develop an application which enables registered users to search and browse through a photo library. Photos should be organized in categories while each photo might be bound to arbitrary categories. Categories are hierarchically ordered. Each photo is described by metadata like EXIF parameters. We choose an interactivity-driven design strategy. The client formulates the following specification: *'The application should be accessible by all usual browsers. The user has to log in. After login he is offered to browse through the categories or search for photos based to the photo's metadata. When the user selects a photo in a category or in the search result, the picture is shown in detail.'*

Figure 3 shows a story space reflecting the given requirements. There are four scenes: the *login* scene for authenticating the user, the *browsing* scene for browsing the categories, the *searching* scene for searching photos and the *viewing scene* for looking at photos in detail.

Even this very small and abstract example reveals important shortcomings for top-down methodologies:

The story space is a valid one but does not reflect the intended application although it reflects the utterances of the client. The client describes runs through the story space, not the story space itself. The client will not explicitly mention the fact that a run may be cancelled or restarted because this is normal behavior. Because the client is usually not able to interpret abstract specifications, top-down approaches force the development team to go through the development steps down to implementation investing huge amounts of resources before the client will notice the error. Experienced modelers will notice this problem very soon. But nevertheless, the outcome of a managed development process has to be independent from personal skills.

It is not possible to implement the given storyboard as a Web application. There are additional scenarios in the story space because every browser offers the possibility to go back in the page history. These scenarios are not visible in the specification but cannot be excluded¹ ('back button problem'). This error is only visible to the experienced modeler or during the tests after implementation.

There are artifacts on every layer of abstraction which are reused between different applications like the *login* scene in the given example. But there are also aspects of the

¹ AJAX may be used to avoid the default behavior of the browser, but if Javascript is turned off, the application will not work at all.

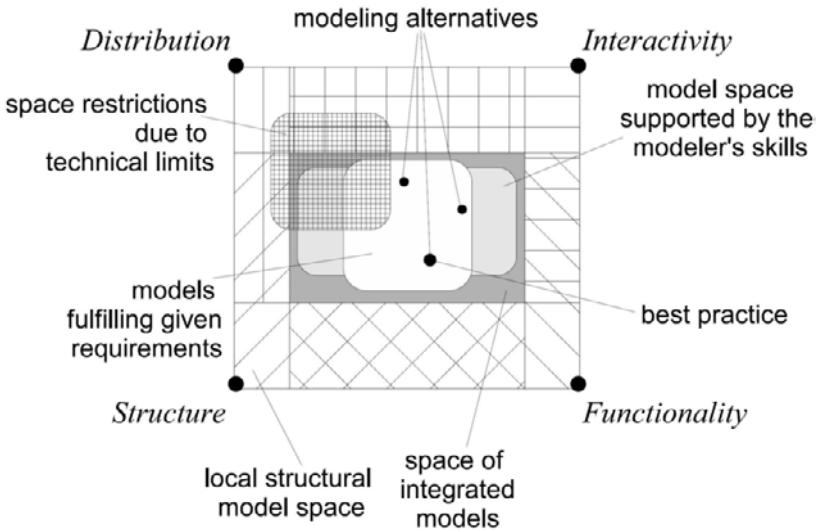


Figure 4. Model Space Restrictions during Co-Design.

application which are orthogonal to the abstraction layers. These aspects are no isolated modelling artifacts but determine the translation of artifacts from one layer to another. Examples are the consistent handling of exceptions, consistent usability handling like internationalization, or the combination of concurrent parts of applications. Usually, there are experiences ('best practices') from former projects how these aspects should be handled. Traditional methodologies assume skillful modelers which are able to perform this transfer of knowledge. Going towards optimizing processes in terms of SPICE requires an explicit handling of this knowledge.

Figure 4 shows the space of possible specifications on an arbitrary layer of abstraction during a Co-Design process. The modelling languages determine the classes of local models for the four aspects of structure, functionality, interactivity, and distribution. The requirements given from the work products of the previous level as well as the skills of the modelers and developers restrict the class of usable models. A further restriction is given by the technical and organizational environment of the intended application. The resulting class of models determines the modelling alternatives for this layer. Each one is a possible solution to the design problem. Some may be considered to be 'good solutions' according to informally given, non-functional parameters.

The use of SPICE in this context encourages to extend every Co-Design based development process in the following fashion to move the outcome of the process towards the 'best practice':

(1) The development process starts on a certain layer of abstraction and climbs down to the implementation layer. If the system is developed from scratch, development starts with the application domain description, otherwise a less abstract layer is chosen.

(2) The development activities are executed on each layer. Development on a certain layer can be considered as a small development process in SPICE. Thus, this process has to be evaluated and its capabilities have to be determined. Because the client is the only stakeholder who is able to evaluate all functional and especially non-functional

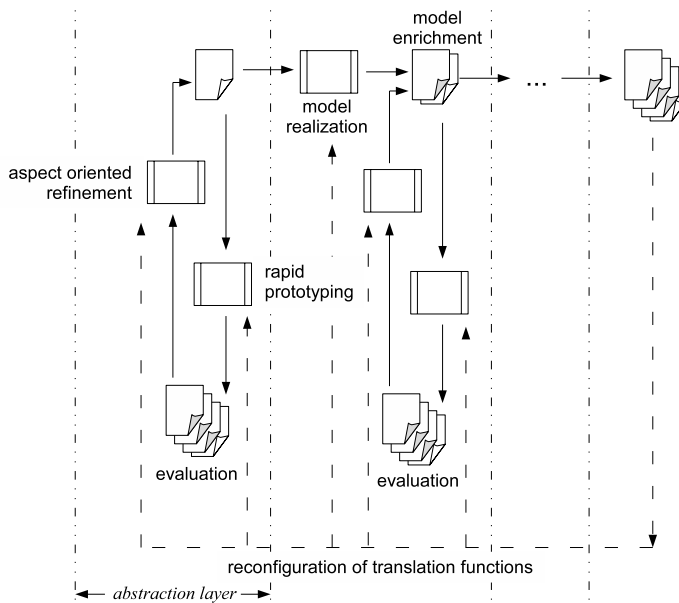


Figure 5. Evaluated Development by Rapid Prototyping and Executable Specifications.

parameters, the assessment has to take place in the client's world. That's why a prototype of the application has to be created. Development of a prototype is a process itself which consumes valuable resources. That's why a mechanism for *rapid prototyping* is needed where prototypes are automatically generated out of the existing specification extended by contextual information representing experiences from former processes.

(3) The capability determination induces a refinement of the outcome of the process. Refinement can be done in two different ways: either manually or by incorporating certain application aspects, e.g., adding transitions to the example from Fig. 3 to reflect cancelled and restarted scenarios. Aspect oriented refinement is only possible if the modelling language supports an appropriate algebra for changing the specification.

(4) If the refined specification has been accepted, it has to be translated to the languages of the subsequent layer ('model realization'). The realization is done by using and extending previously defined mapping functions to incorporate experiences from former projects. The development subprocess for the next layer starts as soon as the translation process is finished.

(5) After the development process reached the implementation layer, the whole process has to be evaluated to adapt the transformation routines for forthcoming developments.

Figure 5 depicts the flow of work products during this development process. The strategy aims at *executable specifications* in terms of a Model Driven Software Development: specifications on higher layers of abstraction are automatically transformed to executable programs by a reference machine without manual transformation steps [FCF⁺06].

The problem of executable specifications on high abstraction levels is simple: specifications on high abstraction levels are simply not executable because important parts of the application are not present. For that reason, the application under development is simulated. The simulated prototype consists of two major components: the specification on a certain level of abstraction and a runtime environment that completes the missing facts from the abstract specification.

The specification is linked with the runtime environment. This results in a specification that is complete at the level of implementation and can be directly executed on the target system. If the abstract specification is refined, it replaces parts of the runtime environment. This process lasts until the specification covers all relevant aspects. Beside being the specification's complement the runtime environment can be used in a second way: creating a production-ready version the runtime environment allows stopping the development process on abstract specification layers.

A golden rule in layered system development is that facts which were modelled at a certain abstraction level are visible in some way in all subsequent specifications. To prevent the 'forgetful' design caused by human interception, the design methodology has to be backed by appropriate system support that guides the human developer through this process and enforces the methodology in a formal way but also on a pure pragmatical level. The system provides different kinds of transformation facilities which can be used during the development process and which are updated in the evaluation phase of each development process:

- Transformers for *rapid prototyping* translate an abstract specification to a specification on the implementation layer such that this specification fulfills the already known requirements as well as all technical and organizational restrictions (it is a valid modelling alternative). Missing parts are filled by defaults, e.g., standard skins for the user interface are generated or data is stored in a database with a derived schema. Transformers are parameterized to allow adaptive prototypes, e.g., different skins for the interface or different data models.
- Transformers for *aspect-oriented refinement* translate specifications to specifications on the same layer of abstraction such that this specification is a valid modelling alternative for a given aspect of the requirements. For example, multiple scenario specifications may be integrated into a multi-modal application.
- Transformers for *model realisations* transform valid modelling alternatives on a certain layer of abstraction to valid modelling alternatives on a layer with lower abstraction. If \mathcal{M}_k is a specification on a layer k and $f_{k \rightarrow I}$ is a transformer which translates \mathcal{M}_k to an executable program, then $(\mathcal{M}_k, f_{k \rightarrow I})$ is called an *executable specification*.

[FCF⁺06] describes the implementation of a generation framework and runtime environment following this approach. Abstract specifications are represented as XML files. The generation framework was implemented in Java and facilitates a dynamically extensible set of transformation classes which transform the specification's XML document, e.g., using the DOM API or XSLT transformations for aspect-oriented refinements (e.g., skinning or application of interaction patterns). Java source code is generated for rapid prototyping and model realisations.

5. Conclusion

Web information systems augment classical information systems by modern web technologies. They aim in supporting a wide variety of users with a large diversity of utilization stories, within different environments and with desires for personal web-enhanced work spaces. The development of WIS is therefore adding the user, story and interaction dimension to information systems development. So far information systems development could concentrate on the development of sophisticated structuring and functionality. Distributed work has already partially been supported. Therefore, these systems have been oriented towards a support for work.

Modern development methodologies must be carefully defined. In order to be used by everybody these methodologies must also be managed in the sense that any development step can be compared with the goals of the development process. WIS development adds another difficulty to this process: continuous change. WIS applications typically evolve and change with the attraction of users and application areas, with the fast evolution of supporting technology and with the upscaling of the systems themselves. Therefore, top-down design is replaced by or integrated with evolutionary design and agile methods. Modern WIS development requires change, revision, reconfiguration on the fly, on demand or on plan.

The Co-Design methodology is one of the methodologies that has been used for the development of classical information systems. Before extending and generalizing the approaches developed for this methodology we carefully assessed the methodology and derived deficiencies of this methodology for WIS engineering. We used an assessment by SPICE that is one of the standards for software process improvement methods. This assessment led to requirements for a fully orchestrated methodology and for development of optimization facilities.

Constant change of systems and continuous change of specifications requires an early execution of any part of the specification. Therefore, we also target in development of supporting technologies that allow to execute changes and to incorporate changes into a running system. The specification is supported by a runtime environment.

References

- [AAMN01] I. Aaen, J. Arent, L. Mathiassen, and O. Ngwenyama. A conceptual MAP of software process improvement. *Scandinavian Journal of Information Systems*, 13:18–101, 2001.
- [Bal98] H. Balzert. *Lehrbuch der Software-Technik*. Spektrum Akademischer Verlag GmbH, Heidelberg, 1998.
- [BS03] Egon Börger and Robert F. Stärk. *Abstract State Machines. A Method for High-Level System Design and Analysis*. Springer, 2003.
- [CMM06] CMMI. Capability maturity model integration, version 1.2. cmmi for development. Technical report, CMU/SEI-2006-TR-008, August 2006.
- [FCF⁺06] G. Fiedler, A. Czerniak, D. Fleischer, H. Rumohr, M. Spindler, and B. Thalheim. Content Warehouses. Preprint 0605, Department of Computer Science, Kiel University, March 2006.
- [FTJ⁺07] Gunar Fiedler, Bernhard Thalheim, Hannu Jaakkola, Timo Mäkinen, and Timo Varkoi. Process Improvement for Web Information Systems Engineering. In *Proceedings of the 7th International SPICE Conference on Process Assessment and Improvement*, pages 1–7. SPICE user group, Korea University Press, Seoul, Korea, 2007.
- [GM97] J. Gremba and C. Myers. The IDEAL model: A practical guide for improvement. *Bridge*, issue three, 1997.

- [HBFV03] G.-J. Houben, P. Barna, F. Frasincar, and R. Vdovjak. HERA: Development of semantic web information systems. In *Third International Conference on Web Engineering – ICWE 2003*, volume 2722 of *LNCS*, pages 529–538. Springer-Verlag, 2003.
- [HSSM⁺04] Brian Henderson-Sellers, Magdy Serour, Tom McBride, Cesar Gonzalez-Perez, and Lorraine Dagher. Process Construction and Customization. *Journal of Universal Computer Science*, 10(4):326–358, 2004.
- [ISO03] ISO/IEC. Information technology – process assessment – part 2: Performing an assessment. IS 15504-2:2003, 2003.
- [ISO04] ISO/IEC. Information technology – process assessment – part 1: Concepts and vocabulary. ISO/IEC 15504-1:2004, 2004.
- [ISO05] ISO/IEC. Information technology – process assessment – part 5: An exemplar process assessment model. FDIS 15504-5:2005, 2005. Not publicly available.
- [KPRR03] G. Kappel, B. Pröll, S. Reich, and W. Retschitzegger, editors. *Web Engineering: Systematische Entwicklung von Web-Anwendungen*. dpunkt, 2003.
- [Kru98] Philippe Kruchten. *The Rational Unified Process – An Introduction*. Addison-Wesley, 1998.
- [Poh96] Klaus Pohl. *Process centered requirements engineering*. J. Wiley and Sons Ltd., 1996.
- [Rol06] C. Rolland. From conceptual modeling to requirements engineering. In *Proc. ER'06*, LNCS 4215, pages 5–11, Berlin, 2006. Springer.
- [RSE04] C. Rolland, C. Salinesi, and A. Etien. Eliciting gaps in requirements change. *Requirements Engineering*, 9:1–15, 2004.
- [Sch92] A.-W. Scheer. *Architektur integrierter Informationssysteme – Grundlagen der Unternehmensmodellierung*. Springer, Berlin, 1992.
- [SO98] S. Saukkonen and M. Oivo. Six step software process improvement method (in Finnish; teollisten ohjelmistoprosessi. ohjelmistoprosessin parantaminen SIPI-menetelmällä). *Tekes 64/98, Teknologia katsaus*, October 1998.
- [SR98] D. Schwabe and G. Rossi. An object oriented approach to web-based application design. *TAPOS*, 4(4):207–225, 1998.
- [TD01] B. Thalheim and A. Düsterhöft. Sitelang: Conceptual modeling of internet sites. In *Proc. ER'01*, volume 2224 of *LNCS*, pages 179–192. Springer, 2001.
- [Tha00] B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.
- [Tha03] Bernhard Thalheim. Co-Design of Structuring, Functionality, Distribution, and Interactivity of Large Information Systems. Technical Report 15/03, Brandenburg University of Technology at Cottbus, 2003.
- [Tha04] Bernhard Thalheim. Co-Design of Structuring, Functionality, Distribution, and Interactivity for Information Systems. In Sven Hartmann and John F. Roddick, editors, *APCCM*, volume 31 of *CRPIT*, pages 3–12. Australian Computer Society, 2004.
- [Tha05] Bernhard Thalheim. Component development and construction for database design. *Data Knowl. Eng.*, 54(1):77–95, 2005.

A Description Logic with Concept Instance Ordering and Top-k Restriction

Veronika VANEKOVÁ^a and Peter VOJTÁŠ^b

^a*Pavol Jozef Šafárik University in Košice*

^b*Charles University in Prague*

Abstract. In this paper we introduce a new “scoring” description logic $s\text{-}\mathcal{EL}(\mathcal{D})$ with concept instance ordering and top-k restriction queries. This enables to create ontologies describing user preferences (ordering of concept instances) and to describe concepts consisting of top-k instances according to this ordering. We construct algorithm for instance problem in $s\text{-}\mathcal{EL}(\mathcal{D})$. The main application is an extension of web modeling languages to model user preferences and top-k in web services and web search.

Introduction

Description Logics (or DLs) [1] denote a family of formalisms used for representing structured knowledge. They provide declarative semantics to express both universal and specific knowledge about the domain of discourse. Every particular DL may differ in the expressive power and in complexity of reasoning. This is very important as we require automatic reasoning tasks like satisfiability checking, inference of implicit knowledge from explicit knowledge base, retrieving instances, etc.

Another important aspect of DLs is that they were chosen as theoretical counterparts of ontology representation languages like OWL [2]. Ontologies are designed mainly to add semantic layer to the current web content. This semantics is machine-understandable and allows better automated processing. Metadata and semantically enhanced applications are the main building blocks of semantic web.

Data on the web is typically large, but incomplete and it originates from heterogeneous sources with different quality. Our knowledge representation system must be able to handle missing values and perform reasoning and query evaluation tasks with huge amounts of data. Under such circumstances, exact user queries would often give too many answers. Therefore we use threshold algorithm as in [7] to restrict the number of results and order the results by their relevance to user query. We introduce a DL that enables such queries and preserves low reasoning complexity.

The paper is organized as follows: Section 1 provides some basic definitions from description logics. Section 2 analyzes the task of retrieving top-k answers within DLs. We provide a formal specification of our approach in Section 3 together with illustrative example. Section 4 contains structural reasoning algorithm to decide instance problem and Section 5 discusses limitations of fuzzy subsumption and proposes a new definition. We describe a description logic based on instance ordering in Section 6. Section 7 brings a short comparison of two related DLs. Finally, Section 8 concludes and presents some problems for further research.

Table 1. Syntax and Semantics of Standard DL Constructors

Abbreviation	Syntax	Semantics
\mathcal{AL}	A	$A^I \subseteq \Delta^I$
	\top	Δ^I
	\perp	\emptyset
	$\forall R.C$	$(\forall R.C)^I = \{a \in \Delta^I \mid \forall b: (a,b) \in R^I \rightarrow b \in C^I\}$
	$C \sqcap D$	$(C \sqcap D)^I = C^I \cap D^I$
	$\neg A$	$(\neg A)^I = \Delta^I \setminus A^I$
	$\exists R. \top$	$(\exists R.C)^I = \{a \in \Delta^I \mid \exists b: (a,b) \in R^I \wedge b \in \top^I\}$
C	$\neg C$	$(\neg C)^I = \Delta^I \setminus C^I$
\mathcal{U}	$C \sqcup D$	$(C \sqcup D)^I = C^I \cup D^I$
\mathcal{E}	$\exists R.C$	$(\exists R.C)^I = \{a \in \Delta^I \mid \exists b: (a,b) \in R^I \wedge b \in C^I\}$

1. Preliminary Theory

This section gives a brief introduction to basics of description logic theory. The language of description logic consists of atomic concept names A, B , role names R, S , bottom \perp and top \top concept symbol, constructors $\sqcap, \neg, \forall, \exists$. More complex concepts are made up according to syntax rules in Table 1. The set of allowed constructors influences the expressivity and complexity of the resulting DL. Depending on the richness of constructors we distinguish several DLs, starting from \mathcal{AL} . Note that $\mathcal{ALC} \equiv \mathcal{AL} \cup \mathcal{E}$.

An interpretation I consists of a domain Δ^I and interpretation of language constructors \bullet^I given by Table 1.

All syntax constructors can be used in TBox assertions of the form $C \sqsubseteq D$ or $C \equiv D$, where C is a concept name and D is arbitrary concept. Thus we assign new names to complex concepts and create our *terminology*. ABox contains *assertions* about individuals. Standard ABox assertions are $a:C$ or $(a,b):R$. An interpretation I is a model of TBox or ABox if it satisfies every assertion. Typical reasoning problems of DLs are described in [1].

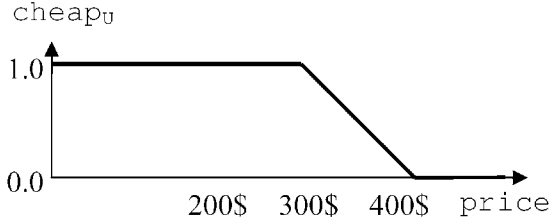
- **Satisfiability:** A concept C is satisfiable with respect to TBox \mathcal{T} if there exists a model I of \mathcal{T} such that C^I is nonempty set.
- **Subsumption:** A concept C is subsumed by a concept D with respect to \mathcal{T} , $C \sqsubseteq_{\mathcal{T}} D$, if $C^I \subseteq D^I$ for every model I of \mathcal{T} .
- **ABox consistency:** ABox \mathcal{A} is consistent w.r.t. TBox \mathcal{T} if there exists a model of both \mathcal{A} and \mathcal{T} .
- **Instance:** individual a is an instance of C w.r.t. ABox \mathcal{A} if $\mathcal{A} \cup \{\neg C(a)\}$ is inconsistent.

2. Supporting Top-k in DL

We start this section with motivation of usefulness of top-k operator. Suppose that our system contains information about notebooks. Many users want to find some cheap notebook with large disk capacity and sufficient memory. However, the idea of “cheapness”, “large disk” or “sufficient memory” will be slightly different for each

Table 2. Notebook Information

id	price	disk	memory
nb1	400\$	90GB	512MB
nb2	200\$	100GB	256MB

**Figure 1.** Fuzzy Predicate cheap_U .

user. Consider a user who specified his requirements as $\text{price} \leq 300$ \$, $\text{disk} \geq 100$ GB, $\text{memory} \geq 512$ MB. If we had following data, no object would exactly fit user's query, but the user would probably regard them as "almost good" and they would still be interesting for him.

Instead of predicates like $\text{price} \leq 300$ \$, we can use fuzzy predicates like cheap . Every object satisfies these predicates to a certain degree from $[0,1]$. For example in case of predicate cheap , every notebook will have a degree of its cheapness. Higher degree means that the user considers this notebook to be cheaper. Thus we gain instance ordering for every attribute (like price). Note that such predicates are user-dependent and we usually denote them as cheap_U , where U is the user's identifier (see Fig. 1). Fuzzy predicate can be viewed as membership predicate of a fuzzy set.

Thus we get the following results:

$$\begin{aligned} \text{cheap}_U(\text{nb1}) &\geq 0,2 & \text{cheap}_U(\text{nb2}) &\geq 1,0 \\ \text{large_disk}_U(\text{nb1}) &\geq 0,9 & \text{large_disk}_U(\text{nb2}) &\geq 1,0 \\ \text{good_memory}_U(\text{nb1}) &\geq 1,0 & \text{good_memory}_U(\text{nb2}) &\geq 0,5 \end{aligned}$$

We still cannot choose the best notebook, because none is better in all attributes. Therefore we use combination function, e.g. a weighted average:

$$\begin{aligned} \text{good_notebook}_U &= @_U(\text{cheap}_U, \text{large_disk}_U, \text{good_memory}_U) = \\ &= (2 * \text{cheap}_U + 2 * \text{large_disk}_U + \text{good_memory}_U) / 5 \\ \text{good_notebook}_U(\text{nb1}) &\geq 0,64 \\ \text{good_notebook}_U(\text{nb2}) &\geq 0,9 \end{aligned}$$

Fuzzy predicates and combination function specify user's requirements; every fuzzy predicate provides ordering of instances (notebooks) according to one property, while combination function $@$ provides overall instance ordering. In the example above, notebook nb2 is clearly better than nb1 for user U because it belongs to concept good_notebook_U with greater degree. Note that combination functions are very expressive; they are generalizations of fuzzy conjunctions and disjunctions. Queries like good_notebook_U presented above are more general than conjunctive or disjunctive queries. We only require that the combination function $@$ is order-preserving in all arguments and that $@^I(1, \dots, 1) = 1$.

Note that fuzziness in this case only models ordering: the bigger is the fuzzy value the better is the notebook. Similar models are scoring or ranking, so fuzzy logic is just one possible formalism. Many valued logic as a formal model provides both proof-theoretic and model-theoretic semantics. Moreover only concepts are touched by this type of fuzziness. There is no fuzziness in roles and hence it would be superfluous to model this phenomenon by fuzzy description logic (see e.g. [10]). This brings us to idea of description logic with ordering on concept instances (either coded by a fuzzy set or any general ordering) and that has top-k operator as an internal constructor of the description logic. For some reasoning tasks, e.g. instance problem or retrieval problem, we may face a combinatorial explosion.

Now we can define $\text{top-k}(\text{good_notebook}_U)$ as a fuzzy set of exactly k objects with the greatest degrees of membership in good_notebook_U . For example

$$\text{top-1}(\text{good_notebook}_U) = \{(nb2, 0, 9)\} \cup \{(x, 0) \mid x \neq nb2\}$$

For a user U_2 with same attribute preferences (cheap_U , etc.) and different combination function:

$$\text{@}_{U_2}(\text{cheap}_U, \text{large_disk}_U, \text{good_memory}_U) = (\text{cheap}_U + \text{large_disk}_U + \text{good_memory}_U) / 3$$

we get:

$$\begin{aligned} \text{good_notebook}_{U_2}(nb1) &\geq 0,7 \\ \text{good_notebook}_{U_2}(nb2) &\geq 0,83 \end{aligned}$$

hence from order theoretic point of view the same ordering as for user U . Nevertheless, for a user with

$$\text{@}_{U_3}(\text{cheap}_U, \text{large_disk}_U, \text{good_memory}_U) = (\text{cheap}_U + \text{large_disk}_U + 3 * \text{good_memory}_U) / 5$$

we get:

$$\begin{aligned} \text{good_notebook}_{U_3}(nb1) &\geq 0,82 \\ \text{good_notebook}_{U_3}(nb2) &\geq 0,7 \end{aligned}$$

If our system contains large amounts of data, this approach would require computing values of all fuzzy predicates for all objects, finding the overall instance ordering and finally selecting k instances with the highest degrees. This process can be optimized with threshold algorithm [7,3]. In the best case, this algorithm needs to process only k objects instead of all objects, but the performance depends on actual dataset. It allows processing data from multiple sources and handles the missing values.

In the following section we present a fuzzy DL that enables top-k queries.

3. \mathcal{EL} with Fuzzy Concrete Domain and Top-k Queries

There has been a considerable effort leading to fuzzy extensions of DLs. More than twenty variants of fuzzy DLs are recently described in scientific papers. Their expressivities vary from simple and effective \mathcal{EL} to very expressive \mathcal{SHIT} , \mathcal{SHOIN} and \mathcal{SROIQ} with yet unknown reasoning algorithms. Reasoning problems are also different; some fuzzy DLs use crisp subsumption, satisfiability and instance problem, some develop fuzzified variants of these problems.

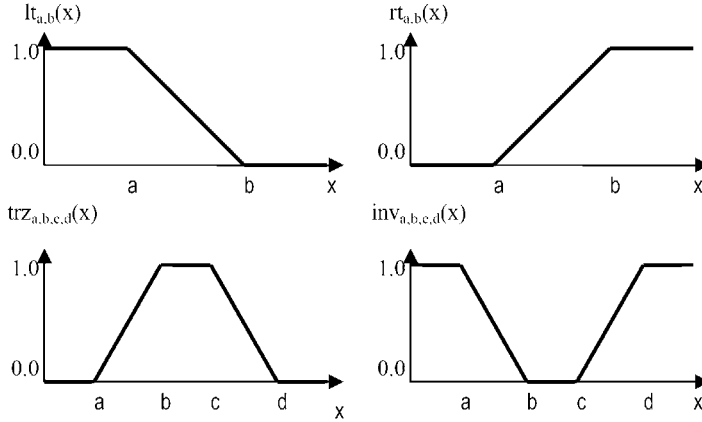


Figure 2. Fuzzy Predicates of Four Basic Trapezoidal Types.

In case of top-k retrieval, we do not need to involve the entire fuzzy theory. In fact we need only the notion of instance ordering to create fuzzy concepts like `large_diskU`, which will include all instances to some degree from $[0,1]$. Fuzzy roles are not necessary. This reflects the intuition that our original data is not fuzzy or uncertain. We know exact disk and memory sizes and prices of notebooks. The source of fuzziness is in user's interpretation of concepts like `large_disk`. Human thinking naturally distinguishes some degrees between “good” and “bad” answers.

Primitive fuzzy concepts have one drawback: they can be interpreted arbitrarily. We want to fix the interpretation of concept like `large_diskU`, so that it would reflect the same user's requirements in any interpretation. This can be achieved with help of concrete domains (see [1], chapter 6). We will use a concrete domain similar to the idea presented in paper [13]. Our concrete domain $D = (\Delta_D, \text{pred}(D))$ has a domain Δ_D of real numbers and a set of unary predicates $\text{pred}(D) = \{\text{lt}(a,b), \text{rt}(a,b), \text{trz}(a,b,c,d), \text{inv}(a,b,c,d)\}$.

Such fuzzy predicates are treated much like fuzzy concepts, but their interpretation is always the same, as seen on Fig. 2.

The choice of constructors is very important when designing a new DL. There is a well-known tradeoff between DL expressive power and reasoning complexity. We include only those constructors that are indispensable for the purpose of top-k: full existential quantification $\exists R.C$ and conjunction $C \sqcap D$. The resulting DL is traditionally called \mathcal{EL} . We further add a concrete domain and a notion of scoring and denote this DL as $s\text{-}\mathcal{EL}(\mathcal{D})$ or scoring \mathcal{EL} with fuzzy concrete domain (also called one sided fuzzy logic because only concepts are fuzzyfied). \mathcal{EL} has very good computational properties as shown in [8,9].

Table 3 shows all syntactic constructors used in $s\text{-}\mathcal{EL}(\mathcal{D})$. Concepts are denoted as C, D , primitive concepts as A , abstract roles as R , concrete roles as S , instances as a, b . The interpretation I consists of nonempty domain Δ^I disjoint with Δ^D and interpretation function \bullet^I . Roles are interpreted as usual $R^I \subseteq \Delta^I \times \Delta^I$, or $S^I \subseteq \Delta^I \times \Delta^D$, i.e. crisp binary relations. However, concepts are not interpreted as crisp subsets of Δ^I , but rather as fuzzy subsets with a membership function $A^I: \Delta^I \rightarrow [0,1]$. The interpretation of $\exists R.C$

Table 3. Syntax Constructors and Interpretations of s- $\mathcal{EL}(\mathcal{D})$

Name	Syntax	Semantics
Primitive Concept	A	$A^I: \Delta^I \rightarrow [0,1]$
Top Concept	\top	$\Delta^I \times \{1\}$
Existential Restriction	$\exists R.C$	$(\exists R.C)^I(a) = \sup \{C^I(b) : (a,b) \in R^I\}$ $(\exists S.P)^I(a) = \sup \{P^I(b) : (a,b) \in S^I\}$
Concept Conjunction	$C \sqcap D$	$(C \sqcap D)^I(a) = \min \{C^I(a), D^I(a)\}$

and $C \sqcap D$ is a slight modification of interpretation in [10], provided that we have only crisp roles and that we use Gödel t-norm for conjunction.

This DL contains fuzzy concepts that represent instance ordering according to some particular attribute. We do not yet have any syntactic constructor for combination functions. It is possible to define $(@ (C_1, \dots, C_n))^I(x) = @ \bullet (C_1^I(x), \dots, C_n^I(x))$ as in [11], where $@ \bullet$ is a fixed interpretation. But this constructor would cause complexity explosion and we will restrict our DL to conjunctions for the sake of simplicity. As we already stated, fuzzy conjunction is a special case of combination function. We will use composition of Gödel conjunctions, $\& \bullet (x, y) = \min \{x, y\}$ which is associative, order preserving and its arbitrary iteration fulfills the condition $@^I(1, \dots, 1) = 1$.

Our knowledge base \mathcal{K} consists of TBox and ABox. TBox contains concept equality axioms $C \equiv D$ and inclusion axioms $C \sqsubseteq D$, where C is a concept name and D is arbitrary concept. ABox contains concept and role assertions about individuals. We use only crisp roles, so the role assertions are the same as usual, i.e. $(a, b):R$, $(a, u):S$ where u is a concrete value. Concept assertions claim that an individual a must belong to concept C , but now we also need to know the exact membership degree of a in C . We define fuzzy ABox assertions as $\langle a:C \geq t \rangle$, where t is a number from $[0,1]$. An interpretation I satisfies (is a model of) the assertion $\langle a,b \rangle:R$ if $(a^I, b^I) \in R^I$. It satisfies the assertion $\langle a:C \geq t \rangle$ if $C^I(a^I) \geq t$. Note that C is a concept name, not a concrete predicate name.

Definition. Let $<_I$ be a total, sharp linear ordering of the domain Δ^I . Let $k \geq 1$. Let $a_1, \dots, a_k \in \Delta^I$ be different individuals such that:

- 1) $C^I(a_1) \geq \dots \geq C^I(a_k)$
- 2) there does not exist any $a_{k+1} \in \Delta^I \setminus \{a_1, \dots, a_k\}$ for which $C^I(a_{k+1}) > C^I(a_k)$
- 3) there does not exist any $a_{k+1} \in \Delta^I \setminus \{a_1, \dots, a_k\}$ for which $C^I(a_{k+1}) = C^I(a_k)$ and $a_{k+1} >_I a_k$ (i.e. a_k is $<_I$ best between ties on the k -th position).

A top- k query is then denoted as top- $k(C)$ with an interpretation:

$$\text{top-}k(C)^I(a) = \begin{cases} C^I(a), & \text{if } a = a_i \text{ for some } i \leq k, \\ 0, & \text{otherwise.} \end{cases}$$

A query top- $k(C)$ is not allowed to appear in our TBox or ABox assertions, it is treated separately. First k members of C^I (with greatest values) remain unchanged and all other individuals will have membership value 0.

Note that the definition above could be ambiguous without condition 3. E.g. if we wanted to choose top-3 objects from $C^I(x) = 0,9$; $C^I(y) = 0,8$; $C^I(z) = 0,7$; $C^I(p) = 0,7$;

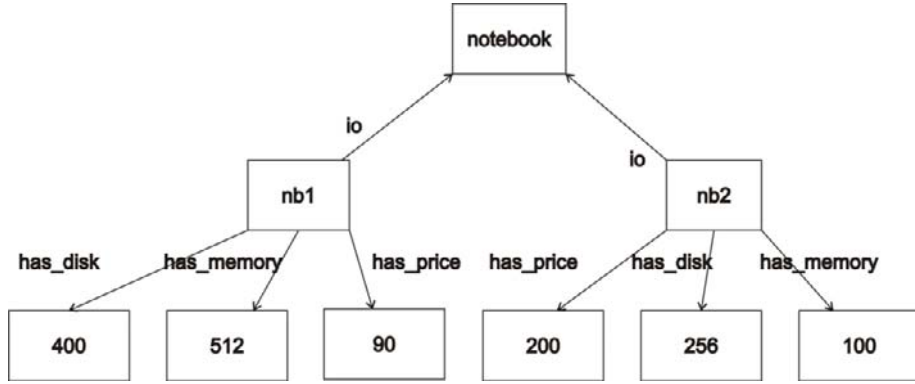


Figure 3. ABox Visualization.

$C^I(q) = 0,6$; we could define the ordering as (x,y,z,p,q) or (x,y,p,z,q) . Thus top-3 objects could be (x,y,z) or (x,y,q) . Therefore we added the notion of total, sharp ordering $<_I$. The real implementation of top-k algorithm can be fully deterministic because sets are in fact represented as sequences in memory or files.

If C^I defines a strict ordering $C^I(a_1) > \dots > C^I(a_k)$, then $\text{top-k}(C)$ is deterministic even without condition 3.

Example. We revise the example from Section 2. Our ABox contains following assertions:

$\langle \text{nb1}:\text{notebook} \geq 1 \rangle$	$\langle \text{nb2}:\text{notebook} \geq 1 \rangle$
$\langle \text{nb1}, 400 \rangle:\text{has_price}$	$\langle \text{nb2}, 200 \rangle:\text{has_price}$
$\langle \text{nb1}, 90 \rangle:\text{has_disk}$	$\langle \text{nb2}, 100 \rangle:\text{has_disk}$
$\langle \text{nb1}, 512 \rangle:\text{has_memory}$	$\langle \text{nb2}, 256 \rangle:\text{has_memory}$

Concrete role names include `has_price`, `has_disk` and `has_memory`. We have only one primitive concept, `notebook`. We use an OWL-style visualization of this ABox on Fig. 3. Nodes of this graph represent concepts, individuals, or concrete values. Vertices represent abstract or concrete roles. Vertices labeled “io” denote “instance of” relationship between an individual and a concept.

User-defined fuzzy predicates will be `cheapU`, `large_diskU`, `good_memoryU`. These sets do not contain notebooks, but rather the concrete values:

$\text{cheap}_U = \text{lt}_{300, 425}$	
$\text{cheap}_U(400) = 0, 2$	$\text{cheap}_U(200) = 1, 0$
$\text{large_disk}_U = \text{rt}_{0, 100}$	
$\text{large_disk}_U(90) = 0, 9$	$\text{large_disk}_U(100) = 1, 0$
$\text{good_memory}_U = \text{rt}_{128, 384}$	
$\text{good_memory}_U(512) = 1, 0$	$\text{good_memory}_U(256) = 0, 5$

Now the user can define his notion of good notebook as a part of TBox:

$$\text{good_notebook}_U \equiv \text{notebook} \sqcap \exists \text{has_price}.\text{cheap}_U \sqcap \exists \text{has_disk}.\text{large_disk}_U \sqcap \exists \text{has_memory}.\text{good_memory}_U$$

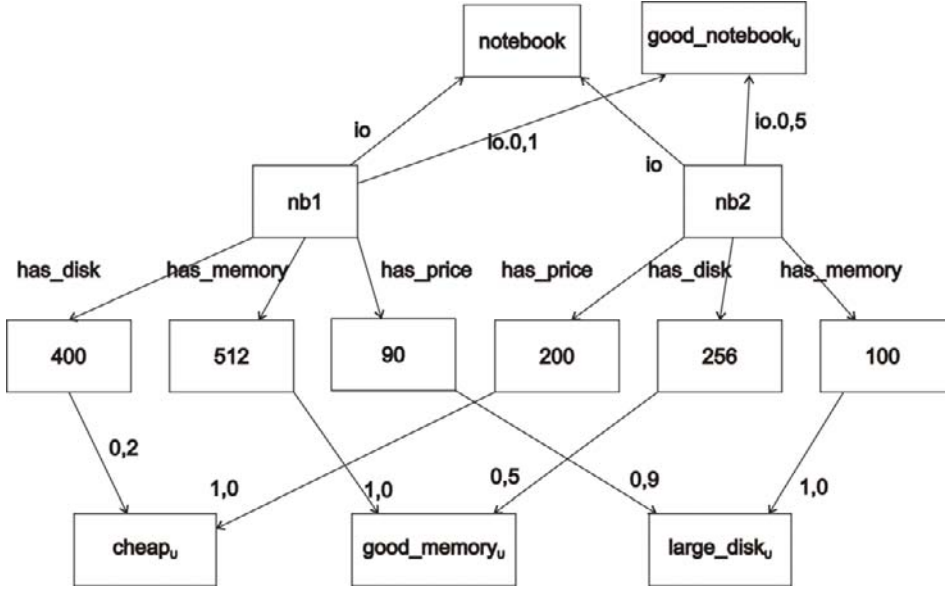


Figure 4. ABox with Inferred Membership in Concept good_notebook_u .

We build an interpretation I that satisfy the following assertions:

$\langle \text{nb1} : \text{good_notebook}_u \geq 0, 1 \rangle$
 $\langle \text{nb2} : \text{good_notebook}_u \geq 0, 5 \rangle$
 $\langle \text{nb1} : \text{top-1}(\text{good_notebook}_u) \geq 0 \rangle$
 $\langle \text{nb2} : \text{top-1}(\text{good_notebook}_u) \geq 0, 5 \rangle$

Note that we gain values different from Section 2 because we used different combination function. The attribute good_memory_u had lower weight than the other attributes, so its influence on the result was less significant.

Although visualization tools do not support fuzzy predicates, we can still illustrate them by adding fuzzy values to vertices. The results can be seen on Fig. 4.

4. Reasoning

Papers about fuzzy DLs do not completely agree in their definitions of reasoning tasks. They usually use crisp subsumption problem defined as follows: a concept C is subsumed by D with respect to a TBox \mathcal{T} , $C \sqsubseteq_{\mathcal{T}} D$, if for every model I of \mathcal{T} and every individual $a \in \Delta^I$ holds $C^I(a) \leq D^I(a)$. However, this condition means that $C^I(a) \rightarrow D^I(a)$ equals 1, where \rightarrow is a fuzzy implication. We can also define *fuzzy subsumption*: C is subsumed by D w.r.t. \mathcal{T} to degree t if for every model I of \mathcal{T} and every individual $a \in \Delta^I$ holds $(C^I(a) \rightarrow D^I(a)) \geq t$. The fuzzy implication \rightarrow is a residuum of fuzzy conjunction.

The crisp subsumption problem $C \sqsubseteq D$ can be decided with the same algorithm as proposed in [12] by finding homomorphism between description graphs for concepts D and C . Some interesting aspects of fuzzy subsumption are discussed in the next section.

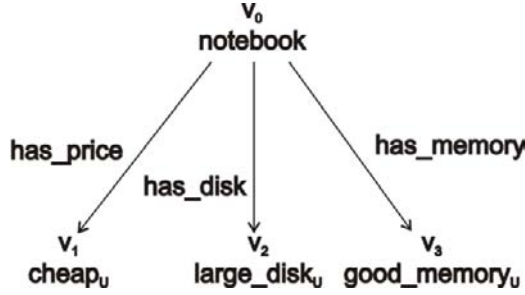


Figure 5. $s\text{-}\mathcal{EL}$ -description Tree for Concept good_notebook_u .

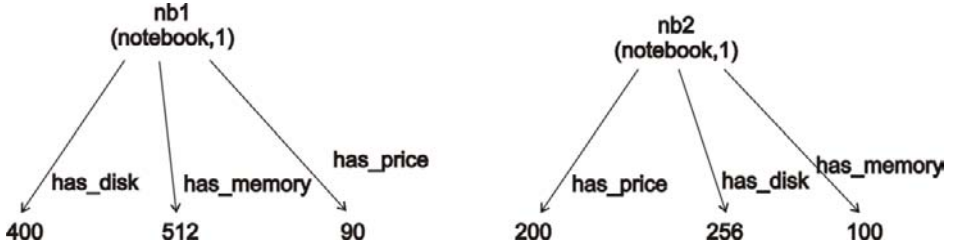
Concept C is *satisfiable* w.r.t. knowledge base \mathcal{K} to degree t if there exists a model I of \mathcal{K} and an individual a such that I satisfies $\langle a:C \geq t \rangle$. For example the concept good_notebook from the previous section is satisfiable to degree 0,5 because we found an individual nb2 which is a member of good_notebook to this degree. Satisfiability to any degree below 0,5 is a consequence because $\langle a:C \geq 0,5 \rangle$ implies $\langle a:C \geq t \rangle$, where $t < 0,5$. We can also define such interpretation that it will be a model of $\langle a:C \geq t \rangle$, $t > 0,5$ (for example an interpretation that sets all truth values to 1). Checking satisfiability of $s\text{-}\mathcal{EL}$ concepts is quite trivial because we do not use negation. It is sufficient to check the syntax of the concept which can be done in linear time.

Instance problem is defined as follows: an individual a is instance of C to degree t w.r.t. \mathcal{K} if for all models I of \mathcal{K} holds $C^I(a^I) \geq t$. Instance problem can be solved with a graph algorithm which is fuzzy modification of [12].

Definition. $s\text{-}\mathcal{EL}$ -description graph is defined as $G = (V, E, I)$. Edges are labeled with role names and vertices (also called nodes) are labeled with sets of concept names with mapping $l: V \rightarrow N_C \times [0,1] \cup \Delta_D$. $s\text{-}\mathcal{EL}$ -description tree is acyclic $s\text{-}\mathcal{EL}$ -description graph where the mapping l is defined as $l: V \rightarrow N_C$.

We can equivalently rewrite any concept C from $s\text{-}\mathcal{EL}$ to the normal form $C \equiv D_1 \sqcap \dots \sqcap D_n \sqcap \exists R_1.C_1 \sqcap \dots \sqcap \exists R_m.C_m$, where each D_i is atomic and each C_j is in the normal form. Concept C can be transformed to an $s\text{-}\mathcal{EL}$ -description tree as follows. Concepts D_1, \dots, D_n will form the label of the root v_0 and for every $\exists R_i.C_i$ we process C_i in the same manner and create a subtree of v_0 . An edge labeled with R_i will connect the root with the new subtree. Figure 5 shows an $s\text{-}\mathcal{EL}$ -description tree constructed for concept $\text{good_notebook}_u \equiv \text{notebook} \sqcap \exists \text{has_price}.\text{cheap}_u \sqcap \exists \text{has_disk}.\text{large_disk}_u \sqcap \exists \text{has_memory}.\text{good_memory}_u$ from the example in Section 3. Vertex v_0 denotes the root.

An ABox \mathcal{A} can be transformed into $s\text{-}\mathcal{EL}$ -description graph. Individual names and concrete values occurring in \mathcal{A} will be vertices of the graph. We have to unfold complex concept assertions, so that our ABox would contain only primitive concept assertions and role assertions. We replace every assertion $\langle a:C \sqcap D \geq t \rangle$ with $\langle a:C \geq t \rangle$ and $\langle a:D \geq t \rangle$ and every assertion $\langle a:\exists R.C \geq t \rangle$ with $\langle a,d \rangle:R$ and $\langle d:C \geq t \rangle$ where d is a new individual. Note that this ABox is equivalent to the original ABox in the sense that it is satisfied with the same set of interpretations.

Figure 6. $s\text{-}\mathcal{EL}$ -description Graph of ABox \mathcal{A} .

For every concrete value u from ABox we create vertex u with the same label. For every individual b used in ABox we create vertex b labeled with $\{(C,t) \mid \langle b:C \geq t \rangle \in A\}$. If the set $\{(C,t) \mid \langle b:C \geq t \rangle \in A\}$ contains both (C,t_1) and (C,t_2) with $t_1 \geq t_2$, we remove (C,t_2) because it carries no additional information. And finally, we create an edge from node a to node b for every role assertion $(a,b):R$ for abstract and concrete roles. Figure 6 shows the result of this process for ABox \mathcal{A} from Section 3. This graph consists of two separate trees; it contains only two individuals and six concrete values, but real knowledge bases would be much larger and more complex.

Definition. Let $H = (V_H, E_H, I_H)$ be a reduced $s\text{-}\mathcal{EL}$ -description graph of ABox \mathcal{A} , $G = (V_G, E_G, I_G)$ be $s\text{-}\mathcal{EL}$ -description tree for a concept C and let a be distinguished individual. A homomorphism φ is a mapping $\varphi: V_G \rightarrow V_H$ such that the following conditions hold:

- for every vertex $v \in V_G$ and for every concept name $C \in I_G(v)$ there exists a tuple $(C,n) \in I_H(\varphi(v))$,
- for every vertex $v \in V_G$ and for every concrete predicate $P \in I_G(v)$ there exists a concrete individual $u \in I_H(\varphi(v))$,
- for all edges $(v,w) \in E_G$ holds $(\varphi(v), \varphi(w)) \in E_H$,
- $\varphi(v_0) = a$.

Let $\&_G$ be infix notation of Gödel conjunction. An *evaluation* is a function $e_\varphi: V_G \rightarrow [0,1]$ defined inductively:

- For every leaf $v \in V_G$, such that $I_G(v) = \{C_1, \dots, C_n\}$, we have unique $(C_1, t_1), \dots, (C_n, t_n) \in I_H(\varphi(v))$. Uniqueness is granted because of the reduction of graph H . We set $e_\varphi(v) = t_1 \&_G \dots \&_G t_n$.
- For every leaf $v \in V_G$, such that $I_G(v) = \{P_1, \dots, P_n\}$, we change the label of $\varphi(v)$ to $(P_1, t_1), \dots, (P_n, t_n)$, where $t_i = P_i(\varphi(v))$. We set $e_\varphi(v) = t_1 \&_G \dots \&_G t_n$.
- For all other $w \in V_G$, $I_G(w) = \{C_1, \dots, C_n\}$, we also have unique $(C_1, t_1), \dots, (C_n, t_n) \in I_H(\varphi(w))$ and successors u_1, \dots, u_m . We set $e_\varphi(w) = t_1 \&_G \dots \&_G t_n \&_G e(u_1) \&_G \dots \&_G e(u_m)$.

If we want to solve instance problem for individual a , concept C and degree t , it is sufficient to create $s\text{-}\mathcal{EL}$ -description tree of C , $s\text{-}\mathcal{EL}$ -description graph of actual ABox and search for homomorphism φ such that $e_\varphi(v_0) \geq t$. As is shown in [12], checking the homomorphism takes polynomial time. We extended this algorithm with unfolding the ABox, reduction of $s\text{-}\mathcal{EL}$ -description graph, changing labels for concrete values and

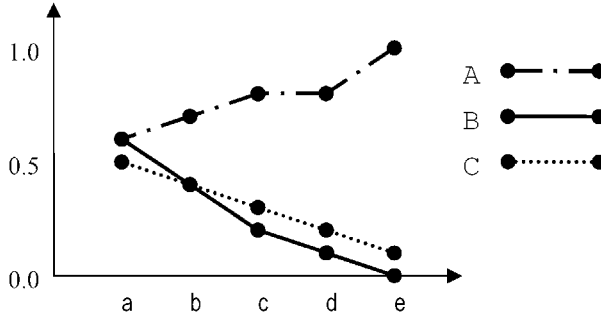


Figure 7. Greatest Degrees of Individuals a,b,c,d,e as Instances of Concepts A, B, C.

checking the condition $e_\phi(v_0) \geq t$ which can be done in polynomial time, so the whole algorithm will be polynomial.

Note that this algorithm finds $e_\phi(v_0)$ which is the greatest degree of individual a being an instance of C w.r.t knowledge base \mathcal{K} . We will denote this value as $C_{\mathcal{K}}(a)$.

We introduce another reasoning problem named *top-k instance problem*: for given concept C and individual a decide if a is an instance of top-k(C) with non-zero degree. We can use a modification of previous algorithm. Instances of C are searched in the *s-EL*-description graph of actual ABox. We remember every pair (b,t) found by the algorithm such that an individual b belongs to C to a degree t. Finally we check if the individual a is among k individuals with the greatest degrees. This approach can be optimized if we use threshold algorithm [3] instead of reasoner.

5. Scoring Approach to Reasoning Tasks

For illustration of order theoretic limitations of fuzzy subsumption, let us consider following example. Assume that we have ABox \mathcal{A} with primitive concepts A, B, C and following assertions (see Fig. 7):

$\langle a : A \geq 0, 6 \rangle, \langle b : A \geq 0, 7 \rangle, \langle c : A \geq 0, 8 \rangle, \langle d : A \geq 0, 8 \rangle, \langle e : A \geq 1 \rangle$
 $\langle a : B \geq 0, 6 \rangle, \langle b : B \geq 0, 4 \rangle, \langle c : B \geq 0, 2 \rangle, \langle e : B \geq 0, 1 \rangle, \langle e : B \geq 0 \rangle$
 $\langle a : C \geq 0, 5 \rangle, \langle b : C \geq 0, 4 \rangle, \langle c : C \geq 0, 3 \rangle, \langle d : C \geq 0, 2 \rangle, \langle e : C \geq 0, 1 \rangle$

Recall definitions of fuzzy and crisp subsumption from previous section: C subsumes D if for every model I and every individual a holds $C^I(a) \leq D^I(a)$ (or $C^I(a) \rightarrow D^I(a) \geq t$ for fuzzy subsumption).

If we have no concept definition (as in this case of primitive concepts), we cannot guarantee that $C^I(a) \rightarrow D^I(a)$ will hold in every model. All models in our example have to fulfill conditions $A^I(a) \geq 0,6$, $A^I(b) \geq 0,7$, $A^I(c) \geq 0,8$, etc. However, we can define such interpretation I that $A^I(a) = 0,6$, $A^I(b) = 0,7$, $A^I(c) = 0,8$, $A^I(d) = 0,9$, $A^I(e) = 1$ and $B^I(x) = 1$ for every $x \in \Delta^I$. Interpretation I is a model of ABox \mathcal{A} (it satisfies all concept assertions), but it is a counterexample against the crisp subsumption $B \sqsubseteq A$. The same principle can be applied for $C \sqsubseteq A$ and for fuzzy subsumptions.

Algorithms for crisp or fuzzy subsumption usually add assertion $(C \sqcap \neg D)(x)$ to ABox with some new individual x , replace complex assertions with equivalent simple assertions and look for contradictions. Description logic $s\text{-}\mathcal{EL}(\mathcal{D})$ does not allow negation. It is possible to construct $s\text{-}\mathcal{EL}$ -description trees G, H for concepts C, D and look for homomorphism from G to H as in [12]. The only difference is the case of concrete predicates. Subsumption between two concrete predicates P, Q can be checked easily because of their fixed interpretation (i.e. we need to check only one interpretation).

To overcome the above problem with fuzzy inclusion, we have to understand fuzziness as scoring. Every concept generates ordering of the abstract domain and every concrete predicate generates ordering of concrete domain. An element with greater membership degree is before elements with lower degree in this ordering.

Order oriented subsumption is denoted $C \sqsubseteq_c D$ or $P \sqsubseteq_c Q$ (c as consistent). It is fulfilled in a structure I if P, Q (or C, D) define the same instance ordering. Note that this condition implies $\text{dom}(C^I) \subseteq \text{dom}(D^I)$.

Definition. Let P, Q be concrete predicate names. P is subsumed by Q , $P \sqsubseteq_c Q$, if $P^I(a) \leq P^I(b)$ implies $Q^I(a) \leq Q^I(b)$ in the fixed interpretation I .

To determine such relationship between arbitrary concepts, we need to choose and consider one interpretation. We unfold complex concept assertion in ABox as in Section 4. If ABox contains both $\langle a:C \geq t_1 \rangle$ and $\langle a:C \geq t_2 \rangle$, $t_1 \geq t_2$, we remove $\langle a:C \geq t_2 \rangle$. Then we define minimal model J as $C^J(a^J) = t$ for every $\langle a:C \geq t \rangle$ and $R^J = \{(a,b) \mid (a,b):R \in \mathcal{A}\}$.

Definition. Let C, D be concept names and J a minimal model of ABox \mathcal{A} . C is subsumed by D , $C \sqsubseteq_c D$, if $C^J(a) \leq C^J(b)$ implies $D^J(a) \leq D^J(b)$.

6. Description Logic with Arbitrary Instance Ordering

As we already stated, every fuzzy concept generates some instance ordering. Conjunctions or aggregation operators are necessary to obtain the final ordering. This principle is similar to decathlon rules: athletes compete in ten disciplines, each discipline is awarded with points according to scoring tables. It is possible to order athletes according to their performance in each discipline. All points are summed up to determine the final order. This is the case when all precise scores are important to determine the final score.

It is also possible to focus on ordering, not on score. This is similar to principles of Formula One: first eight drivers gain points according to the point table, regardless of their exact time or speed. The final ordering is also determined by summing up all points.

Such order-oriented model can be adopted also in fuzzy description logic. Every concept defines linear ordering of the domain. New concepts (orderings) can be constructed from other concepts. Such description logic will be called $o\text{-}\mathcal{DL}^{\text{top-k}}$. In this section we will sometimes write concepts in the infix form $<_c, \leq_c$ to emphasize that every concept is interpreted as ordering.

Language of description logic $o\text{-}\mathcal{DL}^{\text{top-k}}$ consists of usual concept names, role names, individual names and constructors. Instead of bottom and top concept we have a

Table 4. Syntax and Semantics of Standard $\mathcal{O}\text{-}\mathcal{DL}^{\text{top-k}}$ Constructors

Name	Syntax	Semantics
$\mathcal{O}\text{-}\mathcal{EL}^{\text{top-k}}$	A	$A^I \subseteq \Delta^I \times \Delta^I$ is a total linear ordering; if $(a, c) \in A^I$, then c is more preferable than a
	top-k(C)	the ordering on Δ^I consisting of top-k elements of C^I , all other elements are equal to b^I
	G	$G^I = \{(b^I, a) \mid a \in \Delta^I\} \subseteq \Delta^I \times \Delta^I$ is an ordering where all elements are better than bottom element b^I
	W	$W^I = \{(a, t^I) \mid a \in \Delta^I\} \subseteq \Delta^I \times \Delta^I$ is an ordering where all elements are worse than top element t^I
	$@_i(C_1, \dots, C_k)$	as $@_i^I(\leq_1, \dots, \leq_k)$ defined above
	$\exists R.C$	$(a_1, a_2) \in (p\exists R.C)^I$ iff $\exists c_i: (a_i, c_i) \in R^I \wedge (c_1, c_2) \in C^I$, then $(\exists R.C)^I$ is a factorization of $(p\exists R.C)^I$ according to $(p\exists R.C)^I$ equivalence
$\mathcal{O}\text{-}\mathcal{AL}^{\text{top-k}}$	$\neg A$	$(a, c) \in (\neg A)^I$ iff $(c, a) \in A^I$
	$\forall R.C$	$(a_1, a_2) \in (p\forall R.C)^I$ iff $\forall c_i: (a_i, c_i) \in R^I \rightarrow (c_1, c_2) \in C^I$, then $(\forall R.C)^I$ is a factorization of $(p\forall R.C)^I$ according to $(p\forall R.C)^I$ equivalence
\mathcal{C}	$\neg C$	$(a, c) \in (\neg C)^I$ iff $(c, a) \in C^I$
\mathcal{E}	$\exists R.C$	$(a_1, a_2) \in (\exists R.C)^I$ iff $\exists c_i: (a_i, c_i) \in R^I \wedge (c_1, c_2) \in C^I$

concept G for all elements good and W for all elements wrong. In our language we have two specific individuals t for top element and b for bottom element (needed only for interpretation of G and W). We add a new restriction operator top-k and several score combination functions $@_1, \dots, @_n$ with given arity. Complex concepts are constructed according to following rules:

$C \leftarrow$ $A \mid G \mid W$
 $\forall R.C$
 $@_i(C_1, \dots, C_k)$, where $@_i$ has arity k
 $\neg C$
 $\exists R.C$
 $\text{top-k}(C)$

Typical TBox definitions are $C \equiv D$, $C \sqsubseteq D$ and ABox contains concept assertions $(a_1, a_2):C$ and role assertions $(a, c):R$. An interpretation I consists of a domain $\Delta^I = \{b^I, x_1, \dots, x_n, t^I\}$. Every atomic concept C generates a total linear ordering of Δ^I (\leq_C has an associated strict order $<_C$ using trichotomy). Note that brackets are inserted only to visually enclose equal elements in ordering \leq_C :

$$(b^I =_C \dots =_C x_{i1}) <_C (x_{i2} =_C \dots =_C x_{i3}) <_C \dots <_C (x_{im} =_C \dots =_C t^I)$$

For every $@_i$ with arity k and for every k -tuple of orderings $\leq_j \subseteq \Delta^I \times \Delta^I$ aggregation $@_i^I(\leq_1, \dots, \leq_k) \subseteq \Delta^I \times \Delta^I$ is a total linear ordering such that:

- 1) if for every $j = 1, \dots, k$ a $\leq_j c$, then $(a, c) \in @_i^I(\leq_1, \dots, \leq_k)$
- 2) for all $c \in \Delta^I$ $(b^I, c) \in @_i^I(\leq_1, \dots, \leq_k)$
- 3) for all $c \in \Delta^I$ $(c, t^I) \in @_i^I(\leq_1, \dots, \leq_k)$.

Interpretations of language constructors are defined in Table 4. Depending on the richness of constructors we can distinguish several order-oriented DLs from $\mathcal{O}\text{-}\mathcal{EL}^{\text{top-k}}$ to $\mathcal{O}\text{-}\mathcal{ALC}^{\text{top-k}}$. We have to assume that every ordering is total on Δ^I ; otherwise we could not determine the order of some new element or merge two orderings together.

Note that this logic is more general than $s\text{-}\mathcal{EL}(\mathcal{D})$, because fuzzy concept $C^I: \Delta^I \rightarrow [0,1]$ from $s\text{-}\mathcal{EL}(\mathcal{D})$ generates ordering C^I_{\leq} by $(a,b) \in C^I_{\leq}$ iff $C^I(a) \leq C^I(b)$. Also $@_i(C_1, \dots, C_k)$ is more general than k -ary iteration of conjunctors, because it can be defined as a fuzzy aggregation on values of $C^I: \Delta^I \rightarrow [0,1]$. Moreover, note that the definitions of universal and existential restrictions are in concordance with domination theory (e.g. Pareto optimal sets).

7. Related Work

Very similar problem, DL $\mathcal{ALC}(\Sigma)$ with fuzzy sets and aggregation functions (without top- k), is discussed in paper [4]. $\mathcal{ALC}(\Sigma)$ is undecidable and [4] does not provide any explanation concerning the choice of this unprofitable DL. However, all syntactic constructors used in this paper belong to a simpler DL $\mathcal{EL}(\Sigma)$ which is decidable according to [5].

Another simple DL, namely fuzzy DL-Lite, is enhanced with top- k algorithm in [6]. DL-Lite allows only unqualified existential quantification $\exists R$, concept conjunction $C \sqcap D$, basic concept negation $\neg B$ and inverse roles. Top- k queries are answered independently from standard reasoners with a separate algorithm.

8. Conclusions and Future Work

Description logic $s\text{-}\mathcal{EL}(\mathcal{D})$ enables instance ordering according to user preference and subsequent searching for top- k instances. This DL keeps low computational complexity. Every $s\text{-}\mathcal{EL}(\mathcal{D})$ concept is satisfiable. Subsumption and instance problem of $s\text{-}\mathcal{EL}(\mathcal{D})$ (without top- k constructor) can be decided in polynomial time.

From logical point of view, we interpret both concepts and roles as binary predicates. This is a new perspective, unusual for classical DLs. In fact, we cannot replace concepts with functional roles, because classical DLs impose very strict limitation on role constructors and we would not be able to replace new concept definitions from TBox with role definitions. As a part of our future research, we want to explore the possibility of a DL with two sorts of roles and role constructors only (i.e. without concepts). Some roles will represent fuzzy concepts. Another interesting problem is to define reasoning algorithm for $o\text{-}\mathcal{DL}$ to determine subsumption relationship between two orderings.

Acknowledgements

Partially supported by Czech projects 1ET100300419 and 1ET100300517 and Slovak projects VEGA 1/3129/06 and NAZOU.

References

- [1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider: *The Description Logic Handbook*. Cambridge University Press 2003. ISBN 0521781760.

- [2] D. L. McGuinness, F. van Harmelen: *OWL Web Ontology Language – Overview*. W3C Recommendation 2004. URL <http://www.w3.org/TR/owl-features/>.
- [3] P. Gurský: *Towards better semantics in the multifeature querying*. Proceedings of Dateso 2006. ISBN 80-248-1025-5, pp. 63-73.
- [4] S. Agarwal, P. Hitzler: *Modeling Fuzzy Rules with Description Logics*. Proceedings of Workshop on OWL Experiences and Directions, 2005.
- [5] F. Baader, U. Sattler: *Description Logics with Concrete Domains and Aggregation*. Proceedings of the 13th European Conference on Artificial Intelligence (ECAI '98), John Wiley & Sons Ltd, 1998.
- [6] U. Straccia: *Answering Vague Queries in Fuzzy DL-LITE*. Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU-06), 2006.
- [7] R. Fagin, A. Lotem, M. Naor: *Optimal Aggregation Algorithms for Middleware*. Proceedings of the 20th ACM Symposium on Principles of Database Systems, pp. 102-113, 2001.
- [8] S. Brandt: *Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and – what else?* Proceedings of ECAI'04, 2004.
- [9] F. Baader, C. Lutz, and B. Suntisrivaraporn: *Is Tractable Reasoning in Extensions of the Description Logic \mathcal{EL} Useful in Practice?* Proceedings of the Methods for Modalities Workshop (M4M-05), 2005.
- [10] U. Straccia: *Reasoning within fuzzy description logics*. Journal of Artificial Intelligence and Research 14, 2001.
- [11] P. Vojtáš: *\mathcal{EL} description logics with aggregation of user preference concepts*. Information Modelling and Knowledge Bases XVIII, pp. 154-165. IOS Press, 2007.
- [12] R. Küsters, R. Molitor: *Approximating Most Specific Concepts in Description Logics with Existential Restrictions*. Proceedings of the Joint German/Austrian Conference on Artificial Intelligence, pp. 33-47, 2001.
- [13] U. Straccia: *Description Logics with Fuzzy Concrete Domains*. Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-05), 2005.

3C-Drive: New Model for Driver's Auto Evaluation

Juliette BREZILLON^a, Patrick BREZILLON^a and Charles TIJUS^b

^aLIP6, 104 avenue du Président Kennedy, 75016 Paris, France

^bCHArt, 2 rue de Liberté, 93526 Saint-Denis Cedex 02, France

{Juliette.Brezillon, Patrick.Brezillon}@lip6.fr, tijus@univ-paris8.fr

Abstract. Initial training, which concludes by a driving license, is insufficient because new drivers do not know how to contextualize the learned procedures into effective practices. Our goal is to improve the driver's Situation Awareness, i.e. the way in which the driver perceives events in the environment, and the projection of their status in a close future. More precisely, our concern is the way in which the driver evaluates his own competences. Our aim is to make the driver realize his drawbacks and help him to correct them. For this, first, we model drivers' behavior along two approaches, that is, local and global approaches, in order to have a driver model as exhaustive as possible. Second, we educate the driver, helping him to realize his driving's drawbacks. We present in this paper the results of a specific study on driver classification based on the two approaches.

Introduction

Car driving is a complex activity that needs practical experiments to be safe. Initial training ends on a driving license that is often insufficient because the young driver does not know how to contextualize the learned procedures in effective practices. As a consequence, novice drivers are proportionally more involved in accident than experienced drivers [4].

Driver's decision making is not based on an objective state of the world, but on a mental model of the driving task and the conditions in which this task is accomplished. This mental model is a « circumstantial representation » built in a working memory from perceptive information extracted in a scene, and from permanent knowledge stored in the long-term memory. This representation provides a meaningful and self-oriented interpretation of the reality, including anticipations of potential evolutions in the current driving situation. Mental representations are a key element of the driver's cognition. An erroneous representation means, potentially, decision-making errors and unsafe driving actions. [1] illustrate the effect of inexperience at different levels of situation awareness, including information perception, driving situation understanding, and anticipation.

Our aim is to model the driver in the new way by taking in account this mental representation but by linking it to the driver's actions.

Hereafter, the paper is organized as follows. In the first part, we present the context of our work. In the second part, we present the tools we use along each approach (machine learning and cognitive sciences). In the third part, we present our original idea. In the fourth part, we present the methodology we used. In the last part, we present the current state of the project and our results on our driver modeling and we conclude.

1. State of the Art

This section aims to presents classical driver models in the cognitive and the machine learning domains.

1.1. With Cognitive Sciences

GADGET Project. The [9] project, acronym for “Guarding Automobile Drivers through Guidance Education and Technology”, is a European project about road safety. There are the three hierarchical levels – the strategic, tactical and operational level, and a fourth level **is** added concerning “goals for life and skills for living”. The levels also have been divided into three dimensions concerning knowledge/skill, risk increasing factors and self-assessment. The highest level refers to personal motives and tendencies in a broader perspective. This level is based on knowledge like lifestyle, social background, gender, age and other individual preconditions have an influence on attitudes, driving behavior and accident involvement. The idea in the hierarchical representation is that both failures and successes on a level affect the demands on lower levels. Thus driver’s behavior must be analyzed on all these levels and not at the operational level only. We postulate that the discrepancy between the theoretical training, which is validated by the driving license, and the effective training by driving alone (the learning-by-doing) is mainly due to a lack of support in the phase of contextualization of the theoretical training in real life situations, i.e. how to apply effectively general knowledge in a number of specific and particular situations.

Study. [10] shows that it’s better to learn from other people’s errors than from their successes. Two training methodologies were compared and evaluated. One group of person was trained using case studies that depicted incidents containing errors of management with severe consequences in fire-fighting outcomes (error-story training) while a second group was exposed to the same set of case studies except that the case studies depicted the incidents being managed without errors and their consequences (errorless-story training). The results provide some support for the hypothesis that it is better to learn from other people’s errors than from their successes. That’s why we based our driver’s typology on driving’s errors.

Cognitive simulation. Projects likes ARCHISIM [8] and COSMODRIVE [2] and to model traffic simulation by taking in account realistic drivers behaviors. ARCHISIM is a behavioral simulation model and its implementation follows the multi-agent principles. Within ARCHISIM, agents are simulated drivers in virtual vehicles and consist of three subsystems: perception, “interpretation – decision-making” and action. We focus on the “interpretation – decision-making” part. Each agent has a model of its environment and interacts with the other agents (cars, trucks, trams...), the infrastructure (traffic lights) and the road. Each agent has goals and skills.

The objective of COSMODRIVE is to determine whether it is possible to implement reliably a driver model using the techniques from artificial intelligence and based on the theoretical knowledge from cognitive sciences research. This attempt to establish links between different scientific domains, requiring a common tool, is a challenge. A first step of a work that will have to be developed in a long-term time scale, taking into account its quite ambitious objective, is described.

1.2. With Machine Learning

User modeling. In a more general way, machine learning aim to model user behavior in interaction with informatics' systems. Some techniques of personalisation of this interaction have been developed in domains as educative systems [5]. With the development of Internet, these adaptive systems have been apply to Internet to realize some interface adaptation.

Oliver. [11] analyze poor data issues form driver's actions and try to predict driver's next actions. It is to note that moreover her machine learning system, Oliver apply cognitive system to improve the results of the machine learning part.

Dapzol. [6] analyze also poor data issues form driver's actions and try to categorize the driving situation in which the driver is, in order to determine if this situation is quite difficult for the driver.

2. The Tools

2.1. Cognitive Sciences

The method coming from cognitive sciences considered is: Contextual Graphs are a context-based formalism for representing knowledge and reasoning [3]. This formalism allows modeling the different ways in which an individual accomplishes a task. A driving situation represents the different possible scenarios for this « situation solving ». A path in this graph represents a driver's behavior in the driving situation, taking into account the different contexts considered by the user during the situation solving.

2.2. Machine Learning

The process of learning based on the statistical distribution of information in a dataset is used in a class of computational models in cognitive science and psychology to describe human behavior. It is also used in computer science when using data to make predictions. We have selected Hidden Markov Model (HMM), [12] to model driver's behavior. HMM is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. In a hidden Markov model, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by a HMM gives some information about the sequence of states.

3. Presentation of Our Idea

3.1. Limitations of Classical Models

Our first assumption is that classical models of driver's behaviors are limited.

They analyzed a small part of driver's actions and have not a global goal. They only solve sub-problem because of the limitations of the methods. This is due to the

fact that methods used are limited. Cognitive methods are very performed to analyzed very general behaviors but are limited to infer specific behavior, and machine learning methods have the opposite problem.

If we want to take in account all the driver behavior (and not only general behavior or only specific behavior) we have to associate the two methods. That's our second assumption.

3.2. Associate Global Methods and Local Methods

We associate global methods resulting from machine learning and local methods resulting from cognitive sciences. The statistical training aims to model driver's classes whereas the latter relies on a cognitive modeling of drivers' behaviors. The global approach aims to model the driver from numerous data of low level (e.g. movement of eyes when driving). The goal is to process by generalization and abstraction to obtain more conceptual information (e.g. definition of classes of drivers based on real drivers' behaviors). The local approach aims to model each driver at the cognitive level that concerns the highest levels.

The association of the two approaches, the global and local approaches, allows a more complete modeling the driver at all the levels of the matrix proposed in the GADGET methodology. Thus we solve some problem found in literature which are the facts that some studies analyze the driver at one level at the time; for example they studies at the tactical level. But each level depends on highest levels, what is a limit of the other studies. The global methods can only give general information on the driver's behavior in their driving task, whereas there is a high individual variability in this kind of study since each driver is a particular case who acts with a set of contextual cues highly personal that differ one from all other. Moreover, one driver can present very variable behaviors for the same driving task since the contexts in which he is doing this task can be very different.

Thus, a global method constructed from every day life's data allows establishing a driver classification in reality, which have to be completed by a local method. Our classification is very far from the classic "Novice, Medium, Expert", since it's errorsbased and doesn't only take in account the experience of the drivers. This methodology appears justified for us since it shows that classic "Expert" drivers make errors, and that seems to be normal since the "Expert" driver drives a lot and he faces often difficult driving situations. However, in traditional typology, the "Expert" driver is a driver who doesn't make any errors, and that is nonsense according to us.

4. Methodology

4.1. Driver Typology

The first step of our work was to supplement the variables used in the GADGET methodology. We added several kind of variables (as personal and motivational variables, variables dealing with climatic conditions, variables describing the driver's physical or emotional state, variables describing the driving environment and variables describing the car).

All these variables are not GADGET variables, whereas we think they are important for the description of driving activity. For example, the "gender" variable was not

present in descriptions of driving activity; however studies show that women and men do not drive in the same way. All the personal variables that we included are important because they describe the driving experience and the driver's social background, even if they are not directly related to the driving task itself. They were not previously taken into account whereas a person who drives every day for his job, for example, does not drive in the same way as a person who drives only for vacations. Then, we have made a questionnaire of all the variables we have defined and we make our typology based on this questionnaire.

4.2. Driver Typology's Decision Trees

We represent our typology by two decisions trees. The first decision tree represents the driver typology organized by the age of the driver, the second by the driving errors. We represented our typology in this way because of two raisons. We want to classify the driver's "theoretical" errors, according to the age of the driver and to his specific driving skills and we want to classify the driver's "practical" errors, according to his behavior. As we can not instance all errors in the typology, we have selected subgroup of variables that allows us to classify the driver in the typology.

Once we have these two classifications, we can compare them to validate if the driver make awaited errors or not.

4.3. Driving Situation

The second step of our methodology is to choose a driving situation which two characteristics: first, we can be able to define all the scenarios that can happen in this situation; second, all the described behaviors in the typology have to be found in this situation. Once we have defined all the scenarios that can happen, we classify them according to our typology. As an example, if in a scenario a driver had not seen a critical element in the driving situation, then this scenario will be classify as "Not attentive enough" class. This driving situation is used to classify driver's behavior, since it is the link between the driver's behavior and our typology.

4.4. Behavior Detection

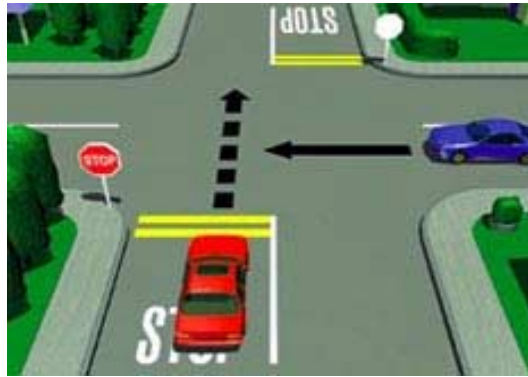
To detect driver's errors in simulation, we used two techniques. We classify the driver thanks to machine learning technique, in the one hand, and thanks to Contextual Graphs in another hand.

By machine learning. They infer started from poor data (as acceleration and braking) some interesting clues to classify driver's behavior.

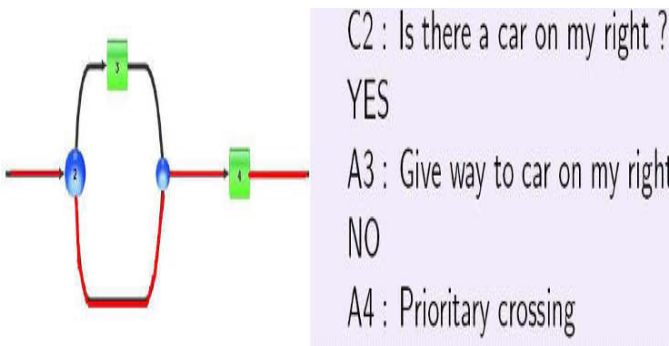
By cognitive Sciences. Contextual Graphs are used to link the scenario and the typology (cf Fig. 1).

4.5. Learning and Validation

Once we have detected driver's errors, we propose him to make some specific scenarios that aim to confront the driver to his drawbacks. For example, if the system has detected that the driver drives too fast, then the system propose him scenarios in which there would be a lot a speed limitation, in order to make uncomfortable the driver.



a)



b)

Figure 1. Example: Suppose that we have two classes of drivers “Respectful of the rules” and “Not respectful of the rules” and that the driver (red car) is in particular crossroad in a. Two possible behaviors: yield the way to the blue car (the way passing by bottom in graph) and pass or pass directly (other way), then if the driver do not yield the way (as the graph show thanks to the way underline in red in b), then we classify the driver in the second class.

After the driver has made the specific scenarios, we validate the learning by proposing him other scenarios, if the system does not detect the same mistakes, the learning is validated.

5. 3C-Drive’s Architecture

To sum up all the methodology, Fig. 2 presents the system architecture. We show our system have three main parts:

Classification: this part aims to classify the unknown driver in our typology (by his age and by his errors).

Learning: this parts aims to determine what are the errors detected and what are the adapted scenarios to correct them.

Validation: this part aims to validate if the learning have work.

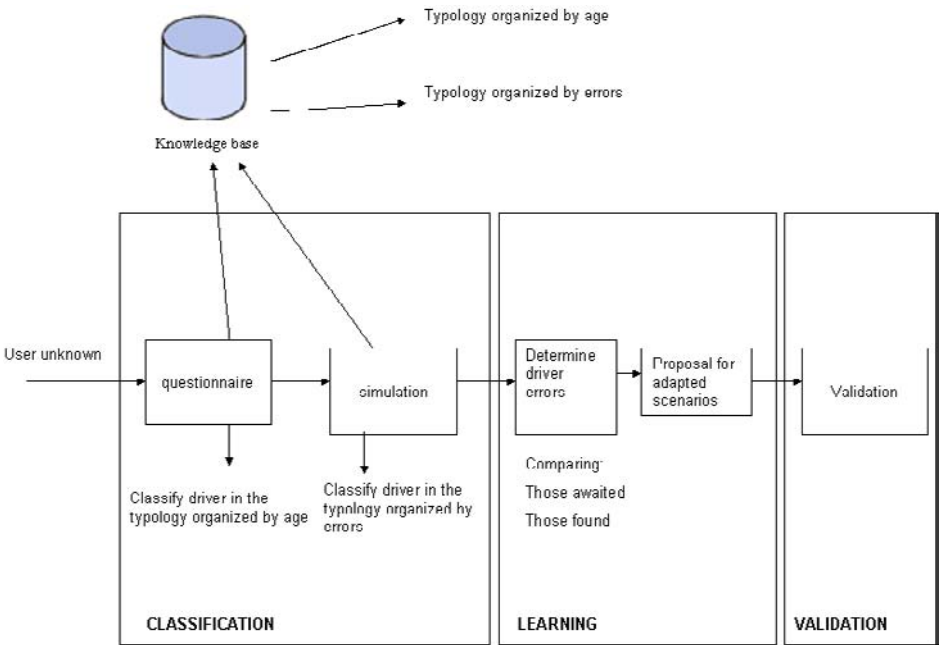


Figure 2. The system architecture.

6. First Results

6.1. Driver Typology

Method. The questionnaire is based on the extended version of the GADGET matrix, concerns 61 variables and has 162 questions. The results are based on 419 relevant answers to that questionnaire. We found 15 classes thanks to hierarchical methods.

We identify for each class the variables that represent the best the class. These variables have a specific value in a class and another value in the others classes. After, we determine in each class the variables that are related to risky behaviors. We then obtain a driver typology that is errors-based. Finally, we analyze driving behavior evolution according to the drivers' age. We wanted to know if young drivers present specific errors different from those of old drivers.

Results. We identify four steps in the evolution of the driving behaviors with the age (see Fig. 3):

Discovering step: it's the step in which drivers discover what driving is, thus errors made at this step concern mainly a lack of competence for driving (as information overload, no evaluation of the necessity of a trip, no respect of the safety margins, etc.)

Risk step: experience coming with driving, the driver looks then for his competences limits by taking risks, thus errors made at this step concern mainly risks (as personal driving style, the no respect to driving rules, etc.)

Stable step: the driver has found and kept his driving style, and the errors made in this step are quite similar to the previous one.

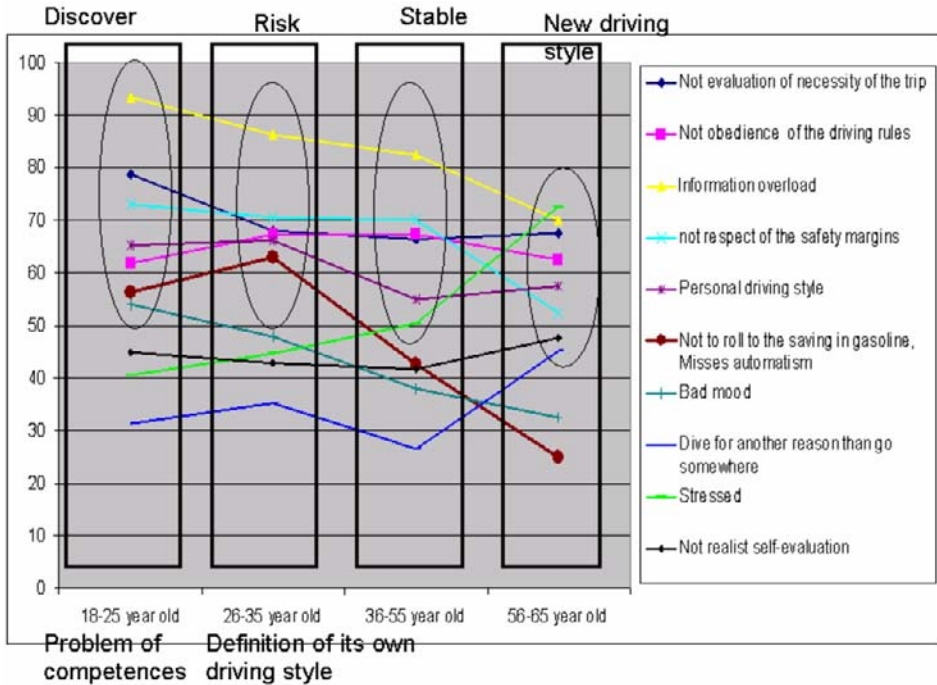


Figure 3. Evolution of the driving behavior among time.

New driving style step: driver's competences decrease with the age; The driver becomes less and less self-confident; The errors made at this step concern the new way to drive (e.g. stressed, not realistic self-evaluation and drive for another reason than go somewhere – which appear at this step).

Figure 3 show that there exist specific errors according to the age of the driver. Young drivers make competence errors by their lack of experience. Later, drivers make risky errors, searching their personal driving style. After, their behavior stays stable. Once older, drivers make errors because there is a shift between their previous way to driving few years ago and the current one. The main problem is a problem of information processing.

6.2. Drivers Typology's Decision Trees

Figures 4 and 5 shows our drivers typology's decisions trees, organized by the age and specific driving skills of the driver (Fig. 4) and by errors (Fig. 5).

6.3. Driving Situation

Consider the real traffic situation—a simple intersection—for which we try to analyze all the driving situations that can happen. We assume only two cars arriving at the intersection. The Highway Code gives the “driver model”: “Let the priority on the car coming from your right.” Because the system must support a given driver, we choose the viewpoint of the driver of car A (coming from the bottom), and analyze all the



Figure 4. Our typology represented by decision tree (age of driver).

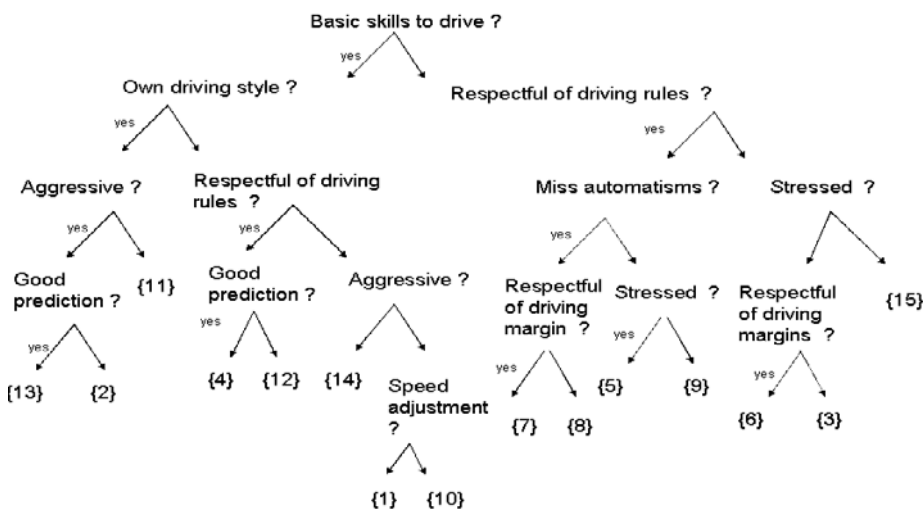


Figure 5. Our typology represented by decision tree (errors of driver).

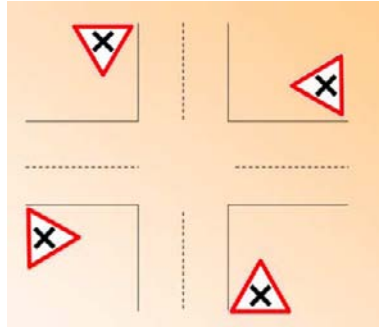


Figure 6. The crossroad.

possible scenarios, first, according from where is coming the car B (from the left, the right or in front of car A), and second, according to the movement of the two cars (turn left, straight ahead, or turn right) at the crossroad (see Fig. 6).

A scenario corresponds to the evolution of a situation (and its context) assimilated to a series of situations.

Driver's behaviors are represented in the context-based formalism called Contextual Graphs that provides a uniform representation of elements of reasoning and of contexts [3]. The principles of contextual-graph building for driving are the following. A contextual graph represents the different behaviors that a driver may present in a given driving situation (an intersection in the following). A path in this graph represents the driver's behavior in that driving situation for given contextual elements (i.e. instantiated elements like "Status" = "In a hurry" and "Weather" = "Rain"). Such contextual elements are instantiated because their value is explicitly considered in the decision making process. A driver can exhibit different behaviors for a given scenario, and the same behavior may appear in different scenarios.

An intelligent support system using such a database can identify the driver's class where is the user, select a situation in which the user has some weaknesses, establish the scenario for which the user may learn how to modify his behavior, follow the driving task realized by the user propose explanations for correcting the user's behavior.

The domain knowledge, which defines driving situations, is represented by contextual elements. A single context of the situation is an instantiation of a (large) part of these contextual elements.

These Contextual Graphs allows us to instance variables in the cognitive way.

6.4. Little Case Study

We take a real traffic situation—a simple crossroad—and try to analyze all the driving situations that can happen. We assume only two cars arriving to the crossroad. We select the viewpoint of the driver of car A (coming from the bottom), and analyze all the options, first, according from where is coming the car B (from the left, the right or in front of car A), and second, according to the movement of the two cars (turn left, straight ahead, or turn right) at the crossroad.

We model all the behaviors by contextual graphs. In the retained traffic situation, each road has a "give way" sign. This means that no priority has been defined and the rule is "priority to the car on the right side".

Since the chosen crossroad has no special priority, the law defines the “theoretical” behavior as “to yield the emerging passage to the vehicles of right-hand side, by having a special vigilance and a deceleration adapted to the announced danger.” There are some restrictions with this panel: the trams have right of way and if the topology of the crossroads obliges it, a special panel added to the first definite the priority. The theoretical behavior established from the law texts is to check that the roadway to cross is free, to circulate with a moderate speed especially if the conditions of visibility are worse, in the event of need, to announce our approach, must engage in an intersection only if our vehicle does not risk to be immobilized in the crossroad area and to anticipate the passage of the vehicles circulating on the other ways. There are two successive parts: the analysis of the situation and the process of the decision making itself.

The possible behaviors

We analyzed what can happen concretely in that crossroad that is not planned by the law. First, the car's driver, which has not the priority, does not stop and enters the crossroad, because for instance, the car's driver thinks that he has time to pass before the other car, or he didn't see it. Then, he can realize that he's making a mistake and decides to stop in the middle of the crossroad. The other car attempts to avoid him. Moreover, the two car's drivers can break down. If a car's driver breakdown, the other car's driver will have to wait until the other starts again and leave the crossroad, or decides to overtake it. If he overtakes, the first car can start again and realize the other car is in front of him and try to avoid him. Or maybe, the other car's driver was not attentive and didn't see that the break down, thus he will have to react at the time he will realize the problem, and he has still some.

Definition of scenarios in our typology

The choice of this case study is the second step of our work. We determine the drawbacks of the driver behavior thanks to this case study. We have several possible scenarios on this situation and each is link to class of our errors-based driver's typology. For example, the driver who is not attentive (and who belongs to the class 12 on our typology) would make the scenario in which he would not see the other car on the crossroad. With the correlation, we are able by making pass this specific driving situation to any driver to identify his drawbacks and his errors in his driving thanks to our typology. By defining scenarios adapted to each class of our typology, we would be able to help any driver to improve his situation awareness.

Conclusion

Driver modeling is an important domain that interests a number of administrations (for a uniform road security in European countries, for the police for interpreting correctly drivers' behaviors, for associations wishing to introduce some changes in the laws, etc.). Our contribution brings at least three new insights on this hot topic. First, we propose a “driver-based” classification of drivers and not an arbitrary classification. Second, we propose an open modeling in the sense that it is possible to incrementally acquire new behaviors of drivers. Third, we use good and bad practices for driver's self-learning, bad practices being mainly used by the system for identifying what is doing a given driver, and how to help him to correct his behavior.

Our goal is now to make an experimentation to validate our hypothesis. This will be done by March, since an experimentation is plan soon, where 50 drivers will have to correct there behaviors.

Acknowledgments

The work presented in this paper is part of the ACC project that is supported by PREDIT and the French Minister of Transportation, concerning mainly the funding of a Ph.D. Thesis.

References

- [1] Bailly, B., Bellet, T., Goupil, C., and Martin, R. (2003), Driver's Mental Representations: Experimental study and training perspectives, at the First International conference on driver behavior and training, Stratford-on-Avon, 2003.
- [2] Bellet, T. and Tattegrain-Veste, H. (2003), COSMODRIVE: un modèle de simulation cognitive du conducteur automobile. In J.C. Spérandio et M. Wolf (eds), *Formalismes de modélisation pour l'analyse du travail et l'ergonomie*. Paris, Presses Universitaires de France, 77-110.
- [3] Brézillon, P. (2005), Task-realization models in Contextual Graphs. In A. Dey, B. Kokinov, D. Leake, and R. Turner (Eds.), *Modeling and Using Context, (CONTEXT-05)*, Springer Verlag, LNCS, 3554, pp. 55-68.
- [4] Chapelon, J. (2005), La sécurité routière en France : Bilan de l'année 2004. Rapport de l'Observatoire national interministériel de sécurité routière (ONSIR), mai 2005.
- [5] Brusilovsky, P. (2001), Adaptive hypermedia, *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110.
- [6] Dapzol N. (2005), Driver's behaviour modelling using the Hidden Markov Model formalism Joung Researcher Seminar of the European Conference of Transport Research Institute.
- [7] Endsley, M.R. (1985), Toward a Theory of Situation Awareness in Complex Systems. *Human Factors*, 37, 32-64.
- [8] Espié, S., (1995), ARCHISIM: Multiactor parallel architecture for traffic simulation, In *Proceedings of: The second word congress on Intelligent Transport Systems'95*, Yokohama.
- [9] GADGET (1999), Formation et évaluation du conducteur, obtention du permis de conduire. Vers une gestion théoriquement fondée du risque routier des jeunes conducteurs. Résultats du projet européen GADGET – Groupe de travail N°3, Stefan SIEGRIST (ed.) Berne 1999.
- [10] Joung, W. and Hesketh, B. (2006), Using "War Stories" to Train for Adaptive Performance: Is it Better to Learn from Error or Success?, *Applied Psychology*, Blackwell Publishing Ltd.
- [11] Oliver, N. (2000), Towards perceptual intelligence: Statistical modeling of human individual and interactive behaviors. Doctoral Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge, MA.
- [12] Rabiner, L.R. (1989), A tutorial on hidden markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2):257-186, February 1989.

The *TIL-Script* Language

Nikola CIPRICH, Marie Duží, Michal KOŠINÁR

*Department of Computer Science FEI, VŠB – Technical University Ostrava,
17. listopadu 15/2172, 708 33, Ostrava - Poruba, Czech Republic*

Abstract. Multi-agent system is a system of autonomous, intelligent but resource-bounded agents. Particular agents have to be able to make decisions on their own, based on the activities of the other agents and events within the system as well as its environment. Thus the agents have to communicate and collaborate with each other as well as with their environment. They communicate by exchanging messages encoded in a near-to-natural language. Traditional first-order languages are not expressive enough to meet the demands of MAS communication. Therefore we make use of the powerful system of Transparent Intensional Logic (TIL). The paper introduces the software variant of TIL, namely the *TIL-Script* functional programming language. We describe its syntax and semantic, as well as the way the TIL-Script facilities are integrated into the whole multi-agent system.

Keywords. Multi-agent system, communication, logical analysis, Transparent Intensional Logic, TIL-Script language.

Introduction

Multi-agent system (MAS) is a system composed of autonomous, intelligent but resource-bounded agents. The agents are active in their perceiving environment and acting in order to achieve their individual as well as collective goals. To this end the agents have to be able to communicate and collaborate with each other as well as with their environment by exchanging messages. As a whole, the system of collaborative agents should be able to deal with the situations that are hardly manageable by an individual agent or a monolithic system. Our project “Logic and Artificial Intelligence for Multi-Agent Systems” is based on the expressive system of Transparent Intensional Logic (TIL). TIL is a higher-order system apt for logical analysis of natural language. Due to its procedural (hyperintensional) semantics, the language of TIL-constructions is ready for implementation. To this end we develop the TIL-Script programming language in which particular messages are encoded and agents’ reasoning is realized.

This paper is an introduction into the philosophy of the TIL-Script language, its role in MAS communication, and the way of implementation as an inference machine of the MAS system.

The paper is organised as follows. Chapter 1 is a description of the architecture of agents’ brain. In Chapter 2 we first briefly describe the basic notions of TIL and their modification into the computational variant of TIL-Script. Then the FIPA standard languages are characterised from the TIL point of view; and finally, we present here an outline of the FIPA compliant implementation of TIL-Script. Concluding Chapter 3 presents a summary of recent work and future research.

1. The architecture of agent's brain

Agent's brain consists of several parts. First, each agent must have a memory, or rather an *Internal Knowledge Base* to store data and knowledge he is "born" with or learns during his life-cycle. Agent's Knowledge Base is available to all the other parts of the brain. Second, in order to implement some intelligence, agents must be equipped with one or more units capable of inferring consequent facts from those stored in the knowledge base. These units make use of various logic systems. The *Prolog Brain Unit*, which is being tested now, makes use of the first-order logic programming in Prolog. The *TIL Brain Unit*, which is under development, makes use of the higher-order expressive system of Transparent Intensional Logic, namely its computational variant, the TIL-Script language. The proposed brain architecture is open to other technologies like JADEX, JESS, etc. In addition to the units that are designed primarily as inference machines, the *Auxiliary Computational Module* provides the functions of mathematical computing (graph algorithms, numeric methods, etc.). The choice of a proper inference unit to make a decision is within the competence of the *Brain Interface*, the communication unit that deals with message accepting, decoding and handling.

A query sent to the brain is encoded in the XML format. The Brain Interface selects an appropriate inference unit to which the XML-query is assigned. The respective unit decodes the query into its internal language, performs relevant inferences, and the resulting answer is returned to a sender *via* the Brain Interface (encoded in the XML format again). The choice of the proper inference unit is made in compliance with the pre-defined map:

$$f: P \rightarrow U,$$

where P corresponds to a set of problems and U is a set of inference units. The XML pack with a query contains also the unique name of the problem. Example of such a map is shown in Table 1.

Table 1. Example of the unit-choosing map.

<i>Problem</i>	<i>Inference Unit</i>
driving	Prolog BU
card playing	Prolog BU
communication	TIL BU

2. Introduction to the TIL-Script language

Though the agents in a multi-agent system are autonomous, they have to communicate with each other in order to achieve their goals. Thus communication is a key topic in the area of MAS. Current FIPA standard languages designed for agents' communication are based on the first-order logic enriched by higher-order constructs whenever the first-order framework is too narrow to meet the demands of a smooth communication. However, these extensions are well-defined syntactically while their semantics is often rather sketchy, which may lead to communication misunderstandings

and inconsistencies. Moreover, the first-order framework is a limiting factor; in particular, agents' attitudes and anaphora processing are a stumbling block of all the first-order theories.

Therefore we design the TIL-Script language based on Transparent Intensional Logic (TIL). Due to its procedural higher-order semantics, TIL-Script is well suited for the use as a *content language* of agent messages. TIL-Script can also serve as a general semantic framework for the known formal languages. Moreover, it is a powerful declarative programming language that can easily be interconnected with the local as well as external knowledge bases of the system.

2.1 Transparent Intensional Logic

Transparent Intensional Logic (TIL) is a logical system founded by Prof. Pavel Tichý.¹ It is a higher-order system primarily designed for the logical analysis of natural language. As an expressive semantic tool it has a great potential to be utilised in artificial intelligence and in general whenever and wherever people need to communicate with the computers in a human-like way. More on the role of logic in artificial intelligence see [7].

Due to its rich and transparent procedural semantics, all the semantically salient features are explicitly present in the language of TIL constructions. It includes explicit intensionalisation and temporalisation, as well as hyper-intensional level of algorithmically structured procedures (known as TIL constructions) which are of particular importance in the logic of attitudes, epistemic logic and the area of knowledge representation and acquisition.

Now we are going to briefly introduce the basic features and main notions of TIL. For details, see also [1], [2], [9] and [10].

2.2 Basic Philosophy

The key notion of TIL is that of a *construction*. It is an algorithmically structured procedure, or instruction on how to obtain an output given some input entities. Constructions are assigned to expressions as their *structured meanings*. There is an analogy between the notion of construction and that of a formula of formal calculi. What is encoded by a particular formula (or a λ -term of the TIL language) is just a construction of the (set of) model(s) of the formula. However, whereas formulas of a formal language are mere sequences of symbols that have to be *interpreted* in order to equip them with meaning, TIL constructions are just those meanings. Particular linguistic terms serve only for their encoding.

2.3 Types of order 1

From the formal point of view, TIL is a partial, hyper-intensional *typed* λ -calculus. Thus all the entities of TIL ontology (including constructions) receive a *type*. Types are collections of member objects. For a type ' α ' we call its members ' α -objects'.

TIL types stem from a type *base*. The base is a (finite) collection of non-empty sets. Every member of the base is an *atomic type* of order 1. For the purposes of natural language analysis, *epistemic base* is used. Its members are specified in Table 2.

¹ See also [9], [10].

Table 2. Epistemic type base.

Notation	Description
\mathbf{o}	The set of truth values: {True, False}
\mathbf{t}	The set of individuals: simple objects — the ‘lowest-level bearers of properties’.
τ	The set of time points. In TIL these are modelled as real numbers. This type is just the set of real numbers.
ω	Logical space, i.e., the set of possible worlds: collections of chronologies of all the logically possible states of the world.

Molecular types are defined as functional closures of atomic types. If $\alpha, \beta_1, \dots, \beta_n$ are types of order 1, the collection of all (including partial) functions/mappings from β_1, \dots, β_n to α is a *type of order 1*: denoted $(\alpha \beta_1, \dots, \beta_n)$.

Next we are going to define types of order n that include constructions. But first, constructions have to be defined.

2.4 Constructions

Constructions are the fundamental building blocks of TIL. Depending on a valuation v , any construction v -constructs an object of some type, or is v -improper (i.e., fails to v -construct anything; such a construction is a ‘blind alley’, an instruction leading to nowhere; remember that TIL is a logic of partial functions). Construction C is an algorithmically structured entity consisting of sub-instructions, i.e. *constituents*, which have to be executed in order to execute the construction C .

There are two kinds of constructions, *atomic* and *molecular*. Atomic constructions are *Trivialisations* and *Variables*; they do not contain any other constituents but themselves. They supply objects on which molecular constructions operate. Molecular constructions are *Composition* (the instruction to *apply* a function to its argument) and *Closure* (the instruction to construct a function by *lambda-abstraction*).² As mentioned above, the formalism of TIL is based on the typed λ -calculus, but it construes the operations of abstraction and application procedurally. Closure (abstraction) is the very procedure of forming a function (and not the resulting function), and Composition (application) is the very procedure of applying a function to an argument (and not the resulting value). Structured meanings are almost exhausted by these two procedures. The functional dependencies underlying compositionality are technically accommodated by means of the interplay between abstraction and application. Note that we strictly distinguish between procedures and their products, and between functions and their values.³

Trivialization: 0a

is an atomic construction constructing the object a without the mediation of any other construction.

Variable: x

is a construction (‘ x ’ is just a name) that constructs an entity depending on valuations— it v -constructs (in the objectual version of Tarskian way).

² There are two other constructions, *Execution* and *Double Execution*, which we are not going to use in this paper.

³ We treat functions as mappings, i.e., set-theoretical objects, unlike the *constructions* of functions.

Composition: $[F C_1 \dots C_n]$

is an instruction of a functional application: If F v -constructs a function f of type $(\alpha \beta_1 \dots \beta_n)$ and each C_i v -constructs an entity c_i of type β_i , the Composition v -constructs the value of f of type α at $\langle c_1, \dots, c_n \rangle$, if any; otherwise the Composition is v -improper.

Closure: $[\lambda x_1 \dots x_n C]$

is an instruction to construct a function f as follows: if variables x_i range over β_i and C v -constructs an object of type α , the Closure v -constructs the following function f : let v' be a valuation that associates x_i with B_i and is identical to v otherwise. Then f is undefined at $\langle B_1, \dots, B_n \rangle$ if C is v' improper, otherwise the value of f at $\langle B_1, \dots, B_n \rangle$ is the entity v' -constructed by C .

2.5 Higher-order types

Each construction is of some order. The order of a construction is the highest order of the types of entities involved in the construction. Basic type of a higher order i ($i \geq 2$) is the type $*_i$ — the collection of all constructions of order i . Molecular types of order i are functional closures of the types of order i , see above.

2.6 Multiagent Systems and Communication

Technologies based on agents are relatively new and very promising. A huge amount of their applications can be found in artificial intelligence and large computer systems. A roadmap of this approach is presented in [6]. In this paper we concentrate on the communication of agents in MAS and particularly on content languages.

Basic standards for communication in MAS are given by FIPA.⁴ According to FIPA standards the basic unit of communication is a *message*. It can be of an arbitrary form but it is supposed to have a structure containing several attributes.

Content of a message is one of the most important attributes. Content carries the semantic meaning of the message. It can be encoded in any suitable language.

The other attributes important from the communication point of view are:

Performative denotes a kind of the message, or rather its communicative act. Basic performatives are:

- *Query*
- *Inform*
- *Request*

Ontology is a vocabulary of the domain specific terms.⁵ These (and only these) terms can be used in the content of the message.

2.7 FIPA SL

One of the objectives of this paper is to propose a new content language for multi-agent systems. But first we are going to briefly discuss the existing standard, the *FIPA SL*.

FIPA SL (Semantic Language) is the only FIPA candidate content language marked as a ‘*standard*’. It is based on the language of the first order logic (FOL) paradigm, but it extends its capabilities. One of the advantages of this approach is that

⁴ The Foundation for Intelligent Physical Agents, see [3], [4], and <http://fipa.org/>

⁵ In the narrow sense of the term ‘ontology’. In general, ontology of a system should specify its stable part, i.e., particular conceptual definitions, rules and necessary general facts.

FOL is a well-known logic, well elaborated and broadly used. But there are disadvantages as well. First, FOL is logic of mathematics; actually it became a shorthand language of mathematics. Its development was motivated mainly by the needs to specify mathematical objects, and FOL is thus appropriate for specifying algebraic structures. But this is definitely not the way the agents communicate, they don't communicate in terms of algebraic structures.

For utilization in multi-agent systems FOL paradigm needs to be extended. FOL formulas express only assertive statements. But queries and requests are valid messages as well. Thus SL defines the so-called *identifying expressions*. Moreover, SL is capable of specifying propositional attitudes of agents to other assertive propositions like “*John believes that it is raining.*” However, the *content* of the embedded-believed clause is conceived to be a *syntactic* object. But it is not a piece of syntax the agent (John) is related to, rather, it is the *meaning* of the embedded clause. The syntactic approach yields difficulties when building a multi-lingual system. Moreover, there are difficulties with iterated attitudes like “John believes that Charles knows that Peter found a parking lot with vacancies”.

The SL language is well-defined from the syntactic point of view. However, there is no proper specification of semantics; one can only consult the section “*Notes on FIPA SL Semantics*” which is (as it says) just notes. The standard counts upon well-known semantics of FOL, but due to numerous extensions it is often not suitably applicable.

This lack of a formal semantics can have unpleasant consequences. There are many objections against syntacticism, the most relevant of which is the problem of translatability. If you are building a multi-lingual system (as it is often needed), the syntactic SL approach makes the transition from one language into another one difficult. Moreover, the syntactic standard can have different interpretations, in particular concerning attitudes, which may yield misunderstandings and inconsistencies in mutual communication of agents.

2.8 The TIL-Script Language

TIL is well-suited for the utilization as a content language in multi-agent systems. Its main advantages are:

Procedural Semantics

Due to the ramified type-hierarchy, constructions are objects *sui generis* of TIL theory, and they are uniquely assigned to expressions as their meanings. Thus we do not have to deal with the problems of syntacticism like those of translation, and with inconsistencies stemming from the need of ‘disquoting’.

High expressibility

The expressive power of TIL is remarkable. Using TIL it is easy to explicitly and adequately analyse almost all the semantic features of natural languages, even those that are the traditional hard-nuts of semantics (like anaphora, *de dicto* vs. *de re* attitudes and modalities).

Its primary purpose

Unlike mathematical logics, TIL has been intended to be a tool for logical analysis of natural language communication. Primarily it has been designed and used for natural languages, but its usage in other areas is straightforward.

But TIL, due to its notation, is not plausible as a computerised content language. That is why we are going to define a computational variant of TIL, namely the TIL-Script language. The reasons are:

1. The notation of the language of constructions is not fully standardized.
2. It is not possible to encode the language of constructions using only ASCII characters. TIL alphabet contains Greek letters, superscripts, subscripts, etc.
3. TIL does not specify an interface to ontologies. Any content language has to be limited to using concepts of a specific domain. These concepts are defined in ontologies.
4. We need one standardized type base. The epistemic type base is not the most convenient for multi-agent systems, because it lacks some very common data types like integers, strings, lists and tuples/sequences. True, the lists and tuples can be defined as molecular, functional types, but for the need of programming simplicity we define them as standard basic types.

Type Base

The type base of the TIL-Script language is an extended epistemic base. To make the language easier to use, we enrich the base with some types common in informatics (see Table 3). The type of actions is currently conceived as a basic one though we are aware of the fact that an action is a complex, molecular entity. The development of logic of actions and events is, however, a subject of further research.

Table 3. TIL-Script Type Base.

<i>TIL-Script Type</i>	<i>TIL Equivalent</i>	<i>Description</i>
Bool	o	The type of truth values.
Indiv	ι	The type of individuals.
Time	τ	The type of time points.
World	ω	The type of possible worlds.
Int		Integer number type.
Real	τ	Real number type.

Molecular Types

Molecular TIL-Script types are collections of functions on pre-defined basic types. For practical reasons, in TIL-Script we also introduce the type of a *sequence*, or more generally of a *list* conceived of as heterogeneous possibly infinite sequence. The following Table 4 is an example of the notation for various molecular types of TIL-Script.

Table 4. TIL-Script Molecular Types.

<i>TIL-Script Type</i>	<i>TIL Equivalent</i>	<i>Description</i>
(Bool Indiv)	(ot)	The type of a set of individuals: characteristic function from individuals to truth values.
(Bool Time Indiv)	(ot _t)	The type of a relation between a real number and an individual; characteristic function from <time, individual> pairs to truth-values.
Indiv@tw	_{t_{ro}}	The type of individuals in intension (individual offices).
List(Indiv)		The type of a list of individuals.
List(Indiv Int)		The type of a list of pairs <individual, number>.
* ₁	* ₁	The type of constructions of order 1.
(Bool Indiv * ₁)@tw	(ot* ₁) _{t_{ro}}	The type of propositional attitudes.

Constructions in TIL-Script

There are six standard kinds of constructions in TIL. All of them have its equivalent in the TIL-Script language, where they are written in nearly the same way as in TIL. The following Table 5 shows the transcription from TIL into TIL-Script.

Table 5. Notation of TIL-Script Constructions.

<i>Description</i>	<i>TIL Notation</i>	<i>TIL-Script Syntax</i>
Trivialization	⁰ C	'C
Variable	<i>x</i>	<i>x</i> or <i>x</i> :Type
Closure (λ -abstraction)	$\lambda x_1 \dots x_n C$	\x1 ... \x2 C
Composition (application)	[<i>C</i> <i>C</i> ₁ ... <i>C</i> _{<i>m</i>}]	[C C1 ... Cm]
Intensional descent	<i>C</i> _{wt}	C@w,t
Execution, Double Execution	¹ C, ² C	^1C, ^2C

Examples.

Now we are going to adduce some examples of natural language sentences analysed in TIL and their transcription into the TIL-Script language.

The *successor function* specification (adding 1).

TIL analysis: $\lambda x [^0 + x ^0 1]$

TIL types: $x/*_1 \rightarrow \tau$; $+/(\tau \tau)$; $1/\tau$;

TIL-Script transcription: `\x: Int ['+' x '1]`

“The president of the Czech Republic is Vaclav Klaus.”

TIL analysis: $\lambda w \lambda t [^0 = \lambda w \lambda t [^0 \text{President_of}_{wt} ^0 CR]_{wt} ^0 \text{VaclavKlaus}]$;

TIL types: $w/*_1 \rightarrow \omega$; $t/*_1 \rightarrow \tau$; $=/(\text{ot})$; *President_of*/(_{t_{ro}}); *CR*, *VaclavKlaus*/_t.

TIL-Script transcription:

`\w \t [' = [\w \t ['President_of@w,t 'CR]]@w,t 'VaclavKlaus].`

2.9 Knowledge and ontologies for TIL-Script

Any content language is strongly related to ontologies. All concepts used or mentioned by a content language must be specified in the system ontology. And vice versa, the content language must be able to use any concept from the ontology.

FIPA definition of ontology is rather vague. It just says that ontology provides a vocabulary of domain specific concepts and relations between them. This leads to diversity in implementations. Usually, ontology takes a frame-like structure, which is well suitable for the FIPA SL language supported by the developer frameworks like Jade.⁶

To specify ontology for TIL-Script, we use either a knowledge base to store the TIL-Script description of particular entity types and names, or the built-in definition of entities, as illustrated by an example below (Table 6). The latest trend is to use the well-established technologies of semantic web. In particular, the OWL language for defining ontologies is well-standardised and defined. However, the OWL support by the implementation tools for multi-agent systems is still a work-in-progress. We are ready to support knowledge extraction from OWL to TIL-Script, which is the subject of the ongoing research.

Table 6. Specification of entities in TIL and TIL-Script.

TIL Notation	TIL-Script Syntax
<i>Prague, Warsaw</i> /t;	Prague/Indiv; Warsaw/Indiv;
<i>Function</i> /(ττ);	Function/(Real Real);
<i>Property</i> /(ot) _{τo} ;	Property/(Bool Indiv)@tw;
<i>Know</i> /(ot* _n) _{τo} ;	Know/(Bool Indiv *)@tw;
<i>C/*_n; Improper</i> /(o* _n);	C/*; Improper/(Bool *);

In TIL you may Trivialize any object, while in the TIL-Script language all the objects that are Trivialized should be specified in the ontology. It may be either a shared system ontology, or an internal ontology of an agent stored in its internal knowledge base. Thus the interface of TIL-Script to the knowledge base or an ontology is realised *via* Trivialization. If an agent does not have the concept in its own knowledge base, he may ask the others, and thus *learn* a new concept. However, it is not possible to trivialize an unknown object, i.e., the object not specified in the used ontology. Moreover, for the use in the TIL-Script language, all the objects as well as ontology classes have to be equipped with TIL-Script types.

Figure 1 below illustrates the mechanism of knowledge exchange between the TIL-Script Engine and the internal knowledge base of agent's brain (which serves as agent's short-term memory), and the communication of the TIL-Script Engine and agent's brain.⁷ The mechanism is realized by a module within the TIL-Script Engine; to

⁶ <http://jade.cse.lt.it/>

⁷ The architecture of agent's brain is described in more details in another paper in this EJC'08 proceedings, namely, Ciprich, Duží, Frydrych, Kohut, Košinár (2008): 'The Architecture of an Intelligent Agent in MAS'.

be more specific, it is the TIL-Core unit that deals with knowledge exchange between agent's memory (e.g., the history of agent's visibility, discourse, messaging...), the external knowledge base (OWL ontologies transformation) and the TIL-Script format.

In standard ontologies, concepts are conceived of as classes of so-called individuals. However, we have to be aware of the fact, that

(a) classes of individuals are not concepts; from the TIL point of view *concepts* are (*closed*) *constructions* of such classes (or of any other entity), and

(b) the so-called 'individuals' of particular ontologies are objects of any (atomic as well as molecular) TIL-Script type of order 1; thus we must not confuse 'ontology individuals' with the TIL-Script elements of the type *Indiv*. For the TIL-Script language this means that any ontology concept (class) whose members are of type α is an object of type $(o\alpha)$, i.e., the set of α -objects. Ontology individuals (members of classes) are directly α -objects, for α being *any* type of order 1.

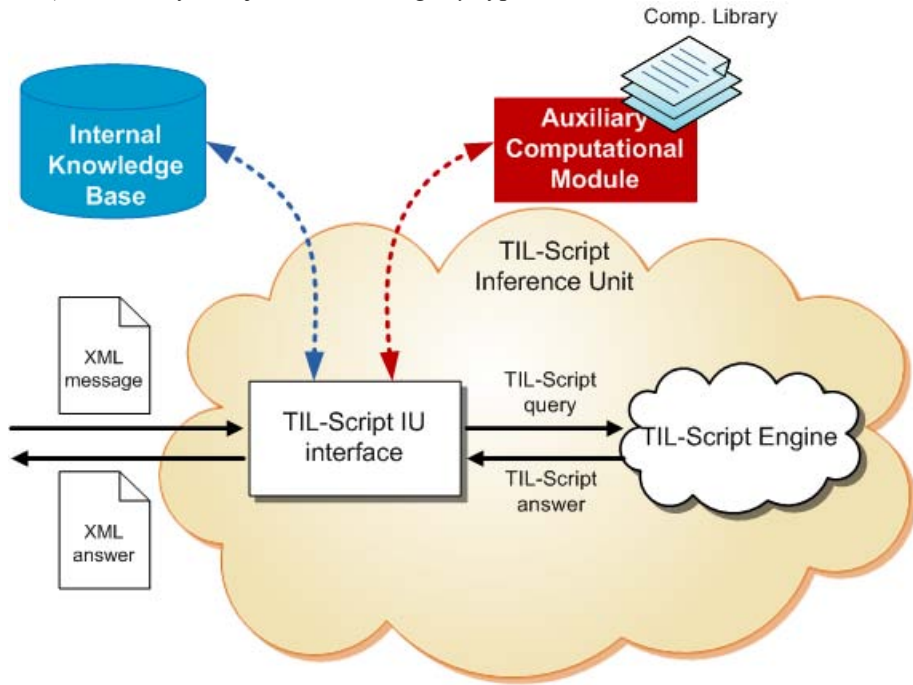


Figure 1. The scheme of the TIL-Script usage.

2.10 Example

By way of conclusion we now present a simple scenario of the communication between two agents using the TIL-Script language.

Scenario

Situation: there are some parking lots and two agents, a driver and a dispatcher:

- *Driver*: an agent driving a car, who wants to park at a suitable parking lot ('suitable for the agent Driver' meaning parking lots in a close distance)
- *Dispatcher*: a dispatcher of the car parking.

The sketch of their communication:

1. *Driver*: I wish you search a suitable parking lot for me.

Now there are two possibilities, 1.1 and 1.2; either the dispatcher understands and answers in the positive or negative, or he does not know which parking lot is suitable for our agent:

1.1 *Dispatcher*: Ok, I've found some parking spaces suitable for you.

Driver: Thank you.

1.2 *Dispatcher*: Sorry, but I don't know what you mean by 'suitable parking lot', please define.

Driver: The most suitable parking lot for me is the one which is nearest to me and is cheaper than \$2.

Ontology

In order to analyse the communication using TIL-Script we need an ontology first. The ontology is presented in Table 7.

Table 7. The ontology needed for this example.

<i>Entity</i>	<i>TIL-Script Type</i>	<i>Description</i>
TheDriver	Indiv	The Driver agent.
TheDispatcher	Indiv	The Dispatcher agent.
TheParking	Indiv	The respective parking lot.
SuitableParkingFor	(List (Indiv))@tw	A list of suitable parking lots (particular individuals obtained from GIS) for agent <i>TheDriver</i>
Refine	(Bool *1)@tw	A request to refine what the used construction means
Seek	(Bool Indiv List (Indiv)@tw)@tw	An action of seeking; arguments: who (agent Driver) and what (list of suitable parking spaces)
Parking	(Bool Indiv)@tw	The property of being a parking lot.
Price	(Real Indiv)@tw	The price of a particular Parking lot
TheLeast	(Real List (Real))	Analytical function which returns the least number of the set of numbers.
TheNearest_to	(Bool Indiv Indiv)@tw	The relation (of being the nearest to).

Communication

Now continuing our example of a human agent requesting a suitable parking space and being asked by the dispatcher to define the meaning of 'suitable parking space', we can specify the communication between the driver and the dispatcher in TIL-Script:

A human agent TheDriver asking for a suitable parking can simply formulate a message for the traffic dispatcher like this:

- "I am looking for a suitable parking lot".

The content of the 'request-type' (FIPA-compliant) message in the TIL-Script is as follows:

- `\w\t['Seek@w,t 'TheDriver
 \w\t['SuitableParkingFor@w,t TheDriver]]`.

The content of the reply message from the dispatcher can be, e.g., as follows:

- `\w\t[['SuitableParkingFor@w,t 'a] = (p1 p2 p3)]`,

which is translated into “The list of suitable parkings for $a = p1, p2, p3$ ”, where $p1, p2, p3$ are particular locations (currently GIS coordinates).

Or, the agent (*The Dispatcher*) may not know what ‘suitable parking for a ’ means. In such a case the content of the reply message is a request for refining:

- `\w\t['Refine@w,t '['\w\t['SuitableParkingFor@w,t
 TheDriver]]]`

translated into “Refine the *concept* of Suitable parking for *TheDriver*”. The agent *Driver* can then specify the parameters of the desired parking.

- `\w\t[x:Indiv['And ['Parking@w,t x] ['And ['< ['TheLeast
 ['Price@w,t x]] '2] ['NearestTo@w,t TheDriver x]]]`

3. Current state of TIL-Script implementation

Currently we develop the first prototype of the TIL-Script language, which is a simplified version of the whole specification. Among its main goals we may list:

- *Simplicity*: The core of the TIL-Script should be kept small and clean. Anything beyond the scope of the basic TIL (special data types, various ontology formats support, etc), should be developed as an additional module.
- *Portability*: The TIL-Script interpret is being developed with the platform independence in mind, so that it is possible to use it on a wide variety of platforms, including IBM compatible systems running Linux or Windows, embedded systems, mobile devices etc. Also portability on programming languages level is taken into account, so that the direct usage of TIL-Script from the most commonly used programming languages is possible.
- *Modularity*: The system is designed to be as modular as possible; for example various syntax dialects can be used just by selecting different parsing modules, or knowledge base storage backend can be changed by switching the *KB* storage module (it is even possible to switch the modules in runtime). Modules are loaded only when needed, and unneeded modules can be removed completely if a limited storage is the concern (i.e. for embedded devices).
- *Scalability*: The design of the interpret makes it possible to exploit extended computer resources, if needed; for instance enterprise grade database backends can be used for large knowledge bases; inference engine can be simply paralleled to exploit multiple CPUs and as the system can run as 64bit application, almost arbitrary amount of memory can be used.

The project itself consists of the following three parts:

1. *Syntactic/Semantic analyzer*

Implements analyzer for parsing input, checking for syntactic validity and type-correctness. Support for various dialects of TIL can be implemented, and used as pluggable modules. As of time of writing, simple analyzer allowing purely prefix

operator notation is implemented, implementation of more advanced analyzer allowing also infix operator notation is now in progress.

2. Inference engine

Implements pluggable modules for inference and reasoning. As the full-fledged TIL logic is undecidable, it is important to design and implement a limited but working inference engine. Three sub-projects on TIL-Script inference engine are currently in progress:

- **TPE:** uses Prolog language interpret for evaluation and inference. This obviously limits the set of solvable problems to a sublanguage of the first-order Horn clause logic; however, it is easy to implement and satisfies the requirements of many applications. This engine is almost finished and implemented.
- **TIM:** implements classical Lambda Calculus paradigms (β -reduction, substitution, etc) augmented with TIL types and constructs. Currently in early stage of implementation.
- **TPS:** rather than an existing project, TPS is an initial research on using algorithms of mixed integer programming for solving logical formulas. Currently in a very early stage; the feasibility of this solution is still a work in progress.

3. Knowledge base

Implements persistent storage for rules and facts. It is implemented as a kind of mapping of TIL-Script knowledge data to relational databases (using relational database backend and SQL for interaction with data) with various data analysis modules.

4. Conclusion

The existing standards for communication in multi-agent systems can be characterised as being syntax driven rather than languages with rigorous semantics. Moreover, they are in principal mostly limited by the expressive power of FOL, which does not meet the goals of a MAS communication. Therefore we proposed the TIL-Script language which is based on the theoretically well-elaborated and highly expressive system of Transparent Intensional Logic. TIL-Script is a semantically driven language suitable for communication in multi-agent systems.

The high expressive power of TIL-Script makes it an appropriate tool to incorporate other logics and languages into its semantic framework, so that TIL-Script can be used as a general specification language. The TIL-Script semantic facilities make it possible to design the communication between humans and computer agents in a smooth and natural way.

The TIL-Script language is being implemented and tested in multi-agent systems using the Python language and the Jadex framework.⁸

⁸ <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/>

Acknowledgements

This research has been supported by the program "Information Society" of the Czech Academy of Sciences, project No. 1ET101940420 "Logic and Artificial Intelligence for multi-agent systems"

References

- [1] Duží, M., Jespersen, B., Müller, J.: Epistemic Closure and Inferable Knowledge. In the Logica Yearbook 2004. Ed. Libor Běhounek, Marta Bílková, Praha:Filosofia, 2005, Vol. 2004, 124-140, Filosofický ústav AV ČR, Praha, ISBN 80-7007-208-3
- [2] Duží, M., Materna, P.: Constructions, <<http://www.phil.muni.cz/fil/logika/til/>>
- [3] FIPA: FIPA SL Content Language Specification [online]. c2002 [cit. 2007-11-11]. Available from WWW: <<http://www.fipa.org/specs/fipa00008/>>
- [4] FIPA: FIPA Abstract Architecture Specification [online]. c2002 [cit. 2007-11-11]. Available from WWW: <<http://www.fipa.org/specs/fipa00008/>>
- [5] Kohut, O.: Brain for agents in multi-agent systems, In WOFEX 2007, Faculty of electrical engineering and computer science, VŠB – Technical University of Ostrava, 2007, p. 291-296
- [6] Luck, M., McBurney, P., Shehory, O., Willmott, S.: Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing. University of Southampton on behalf of AgentLink III, 2005
- [7] Thomason, R.: Logic and Artificial Intelligence[online]. The Stanford Encyclopedia of Philosophy, Available from WWW: <<http://plato.stanford.edu/archives/sum2005/entries/logic-ai/>>
- [8] TIL-Script Home Page [online]. c2005 [cit. 2007-11-11]. Available from WWW: <<http://www.cs.vsb.cz/til/>>
- [9] Tichý, P.: *The Foundations of Frege's Logic*. Berlin, New York: deGruyter, 1988.
- [10] Tichý, P.: *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, and C. Cheyne (eds.), Prague: Filosofia; Dunedin: University of Otago Press, 2004.

An Efficient Method for Quick Construction of Web Services

Hao HAN, Yohei KOTAKE and Takehiro TOKUDA

{han, kotake, tokuda}@tt.cs.titech.ac.jp

*Department of Computer Science, Tokyo Institute of Technology
Meguro, Tokyo 152-8552, Japan*

Abstract. With the development of the Internet, Web services, such as Google Maps API and YouTube Data API, become more important and convenient for the Web knowledge distribution and integration. However, currently, most existing Web sites do not provide Web services. In this paper, we present a partial information extraction technology to construct the Web services from the general Web applications quickly and easily. Our implementation shows that our approach is applicable to different kinds of Web applications.

Keywords. Web application, Web service, information extraction

Introduction

Web services are XML-based software systems designed to support interoperable machine-to-machine interaction over networks, and executed on remote systems hosting the requested services. By using Web services, Web sites can publish their functions or messages to the rest of the world. Many Web sites, such as Google, YouTube and Flickr, provide their Web services like online maps, videos and pictures, and these Web services are widely used for Web information integration and popular with the mashup system development.

However, unfortunately, most existing Web sites do not provide Web services. For the Web sites, Web applications are still the main methods for the information distribution by the Web documents in a standard format supported by common browsers such as HTML and XHTML. For example, CNN [1] lets users search for the online news by inputting the keywords at Web page. Once the users submit the search, CNN would present the news search results. However, this news search service can not be integrated because CNN does not open this application by a Web service. Similarly, Wikipedia does not provide the official Web service APIs and it is difficult for the developers to integrate it with other Web services.

In this paper, we propose a partial information extraction approach to construct the Web services. We design the extraction patterns for the target Web sites, and use these patterns to extract the partial information from Web documents to create the resulting tables, finally respond to the requests of users with corresponding field values like the real Web services. We run the actual extraction and query processes of the constructed Web services at a proxy server, and provide the designed Web service interfaces.

The organization of the rest of this paper is as follows. In Section 1 we give an overview of our Web service construction approach. In Section 2 and Section 3, we explain the method of partial information extraction, information query and interface configuration in detail. We give the implementation and evaluation of our approach in section 4. In Section 5 we give related work, and finally conclude in Section 6.

1. Overview

A real Web service responds to the requests from users by returning the data from server-side *information resources*, usually a database. For our Web service construction approach, the *information resources* are the Web applications. We need to extract the information from the Web applications to generate the resulting tables like the tables of database.

We give an overview of our Web service construction approach. Our approach is based on the *partial-information-extraction-method* and *resulting-table-query-method* as described in the following steps:

Firstly, we select the target parts from Web pages to generate the extraction patterns, which comprise the names and data types for the selected parts.

Secondly, we run the constructed Web services at a proxy server, and configure Web service interfaces for them.

Thirdly, the Web services use the extraction patterns to extract the partial information statically or dynamically according to the requests of users to create the resulting tables.

Finally, the Web services search for the desired information from the resulting tables and respond to the users with the corresponding field values.

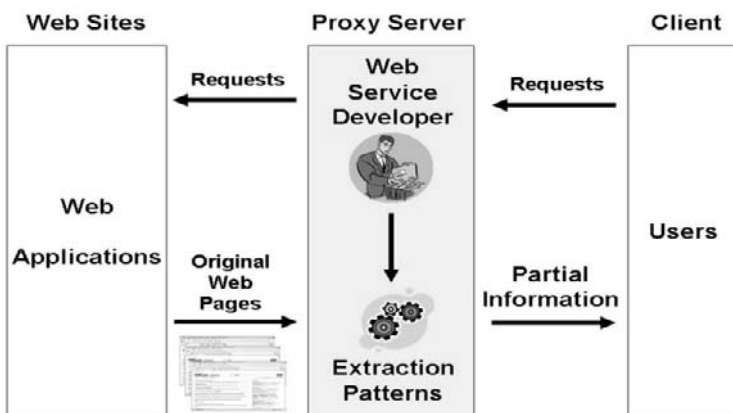


Figure 1. Outline of our approach

We explain our approach in detail in the following sections.

2. Web Information Extraction Method

Not all the contents of Web applications are necessary and useful for Web service construction. As the Web service developers, we need to select the available information from the Web pages according to the actual function of the constructed Web service. For example, we just select the result list from a search result page except the parts such as advertisements if we want to construct a Web service to realize the search function.

We design the extraction pattern for each target Web application and use the extraction pattern to acquire our selected partial information statically or dynamically. Each HTML document of Web application can be parsed as a tree structure, and our extraction method is based on the analysis of the tree structure.

2.1. Extraction Pattern

Our extraction pattern comprises two parts: data type and path. Data type represents "What kind of information is needed?" and path represents "Which part of document is needed?". We define a field name and data type for each target part, then acquire the path of the part.

2.1.1. Data Types Definition

There are many kinds of information in Web applications such as a piece of text or a group of photos. In order to extract information accurately and select the target parts conveniently, we define a name for each target part and its data type for the extraction of partial information. The data type includes two kinds of information: property and structure.

Property is text, object or link. Text is the character string in Web pages such as an article. Object is one instance of the photo, video and other multimedia file. Link is a reference in a hypertext document to another document or other resource.

Structure is single occurrence or continuous occurrence. A single occurrence is a node without similar sibling nodes such as the title of an article, and the continuous occurrence is a list of nodes with similar paths such as result list in a search result page.

There are six kinds of data types: single text, continuous text, single object, continuous object, single link and continuous link. For example, for a news article Web page shown in Fig. 2, the news title is a single text with name *Title*, the news contents are continuous text with name *Paragraph*, one photo is a single object with name *Photo* and the related news are shown as continuous link with name *Related Link*.

2.1.2. Parts Selection

We select the target parts to reach the partial information. Each target part is represented by a node, and each node can be represented by its path from the root. We use the following form to save the path of a selected part:

$body : 0 : ID/N_1 : O_1 : ID_1/N_2 : O_2 : ID_2/.../N_{n-1} : O_{n-1} : ID_{n-1}/N_n : O_n : ID_n$

Where, N_n is the node name of the n -th node, O_n is the order of the n -th node among the sibling nodes, ID_n is the ID value of the n -th node, and N_{n-1} is the parent node of N_n .



Figure 2. Data types

For a group of parts whose structures are continuous occurrence, we do not need to select all the parts one by one and just select one of them. For example, in a search result page containing ten result items, we just select the first item and define the structure as continuous occurrence for it.

2.2. Partial Information Extraction

We use the extraction pattern that contains defined data types and acquired paths to realize the partial information extraction.

2.2.1. Similar Paths

For a Web site, the response Web pages are similar to each other in common if the requests are similar and sent to the same target because there are a number of Web pages dynamically generated by the same server-side programs. For example, in the BBC Country Profiles site [2], there exists a collection of 200 or more country/region information including most recent basic information such as capital city, population and area information. If we choose a country, the Web page of this country would be returned, and all the country profile pages are similar to each other as shown in Fig. 3. So, the paths of the similar parts of these country profile pages are similar to each other, too.

Similar Path of Part: Two paths are similar to each other, if these two paths have the same forms ignoring the difference of orders of nodes among sibling nodes, and the difference of orders is within a defined deviation range. The form of path is as follows:

$$body : 0 : ID/N_1 : (O_1 - h \sim O_1 + h) : ID_1/N_2 : (O_2 - h \sim O_2 + h) : ID_2/\dots/N_{n-1} : (O_{n-1} - h \sim O_{n-1} + h) : ID_{n-1}/N_n : (O_n - h \sim O_n + h) : ID_n$$

Where, N_n is the node name of the n -th node, O_n is the order of the n -th node among the sibling nodes, ID_n is the ID value of the n -th node, N_{n-1} is the parent node of N_n , and h is the deviation value. An example of similar paths is shown in Fig. 4.

We use the ID value to choose the most appropriate paths with the minimum deviation value from the deviation range.

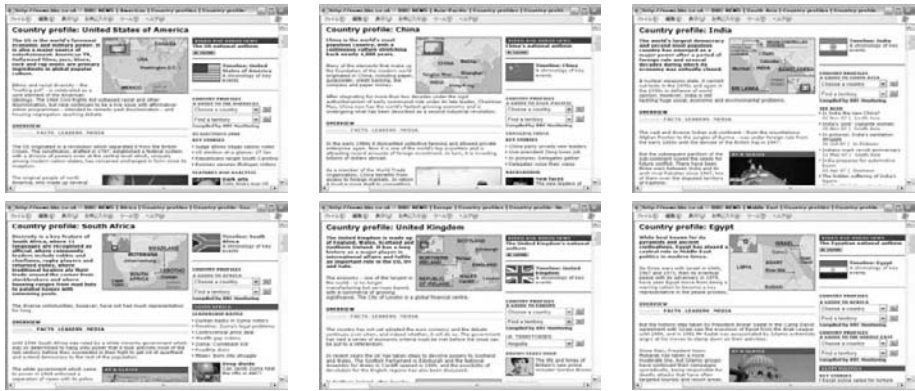


Figure 3. All the BBC country profile pages are similar to each other

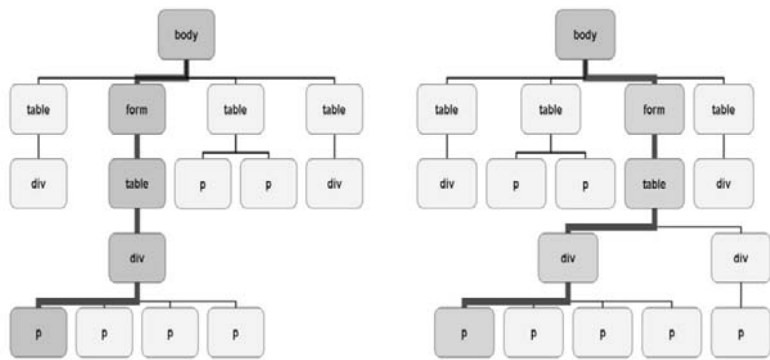


Figure 4. Similar paths

Although the layout of Web pages is relatively steady and not changed randomly usually, some Web sites change the layout of Web pages periodically or irregularly. The similar paths could find the correct target parts automatically after the Web sites change the layout of Web pages if the changed paths of target parts are in the deviation ranges.

2.2.2. Node List Extraction

In the tree structure of HTML document, each path represents a node. We extract the nodes according to the corresponding paths. If the data type of a part is continuous occurrence, we use the following steps to extract the list of nodes:

1. We use the path to find the corresponding node N .
2. We get the parent node P of N .
3. We get the subtree S whose root node is P .
4. We get the node list L of which each has the same path as N without considering the orders of child nodes of P under S .
5. If we find two or more than two nodes in L and these nodes are different from the nodes of other selected parts, or P is $\langle \text{body} \rangle$, then L is the final node list. Otherwise, we set the parent node of P as the root node of S , then go to Step 4.

Each node of the extracted node list L represents a part of continuous parts as shown in Fig. 5.

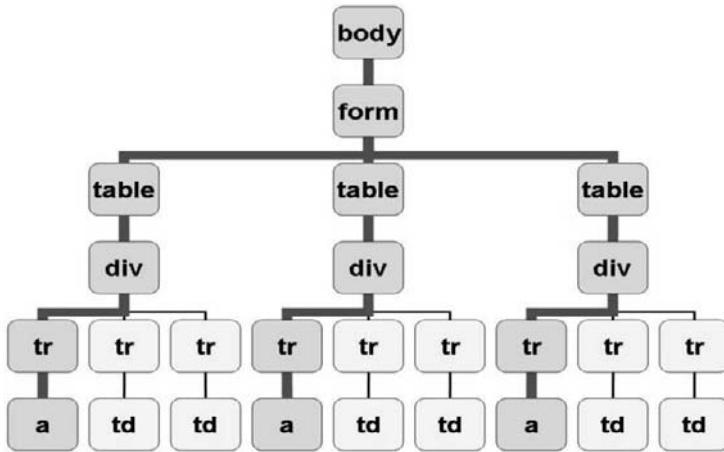


Figure 5. Node list extraction

The table is a kind of often-used information for Web service construction. There are two types of tables in Web pages: horizontal type and vertical type. Each column of a horizontal type table is usually identified by a column name and the first row is the header row to display the column names. Each row of a vertical type table is usually identified by a row name and the first column is the header column to display the row names. If the users want to construct a Web service based on a table in a target Web page, they need to select the first cell of each column of a horizontal table or the first cell of each row of a vertical table, and set the structure as *continuous occurrence* for them. Fig. 6 shows the brief steps of node list extraction from tables.

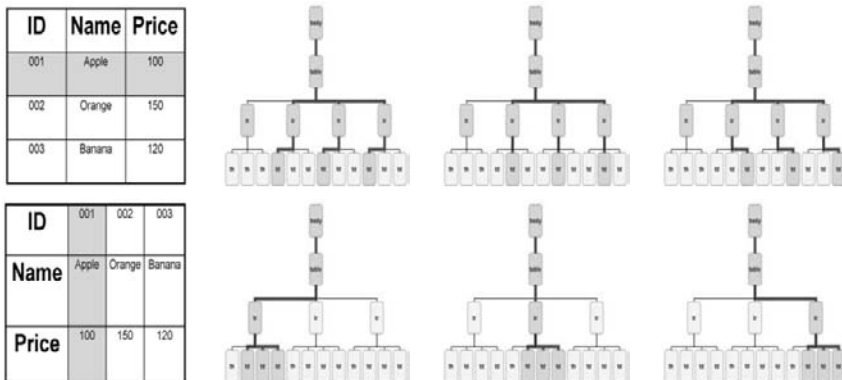


Figure 6. Parts selection and nodes extraction from tables

2.2.3. Extraction Result

According to the defined data types, we extract the partial information from the extracted nodes in text format excluding the tags of HTML document as described in Table 1.

Table 1. Partial information extraction

Data Type	Partial Information
single text	node value of corresponding single leaf node
single object	attribute value of corresponding single node
single link	embedded link value of corresponding single node
continuous text	leaf node values of corresponding list of nodes
continuous object	attribute values of corresponding list of nodes
continuous link	link values of the list of nodes

For example, the extracted information of a photo is the value of attribute *src* of node ``, and the extracted information of a link is the value of attribute *href* of node `<a>`.

Fig. 7 shows the extraction result of Yahoo Finance [3]. The extracted partial information is a document in XML format. The node name is the defined name of target part and the node value is the extracted information.

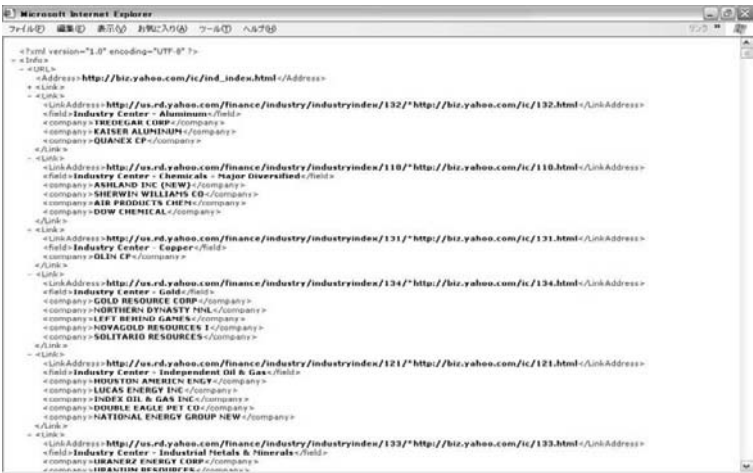


Figure 7. Extraction result in XML format

3. Web Service Construction and Configuration

We run our constructed Web services at a proxy server. In the proxy server, the extraction patterns are used to extract the partial information and the resulting tables are generated. By a designed interface, the users send the requests to proxy server and get the responses from the resulting tables.

3.1. Extraction Frequency and Mode

Usually, for the users of a real Web application that is suitable for Web service construction, there are two basic types of methods to send their requests. The first type is to enter a keyword into an input field by keyboard and click the submit button by mouse to send the query. The second type is to click a link or an option in drop-down list in Web page by mouse to view a new Web page. For the first type, the target Web pages are generated dynamically in server-side. The URLs of target Web pages are changed by the requests of users usually, and we call them *Dynamic URLs*. We use HtmlUnit [4] to emulate the submit processes after the users send the requests and get the target Web pages. For the second type, the URLs are not changed by the requests of users. We call them *Static URLs*.

In the Web applications, some information is static or changed periodically. We do not need to extract them dynamically, especially the static information with static URLs, after the users send the requests every time. We define three kinds of extraction frequencies for Web services: dynamic extraction, periodic extraction and static extraction.

1. *Dynamic Extraction*: The Web services extract the partial information dynamically after the users send the requests. It is suitable for the information extraction from real time system or the search result Web pages with dynamic URLs such as CNN news search result pages.
2. *Periodic Extraction*: The Web services update the resulting table by extracting the partial information at a regular interval such as one hour, one day or one week. For example, the weekly ranking of hot items is updated once a week at Rakuten [5].
3. *Static Extraction*: The extracted information is unchanged in a long period such as the basic information of a country/region. For example, the capital city and area information of most countries are unchanged and the population and life expectancy information remain unchangeable in a long period at BBC Country Profiles.

We can set the extraction frequencies for the constructed Web services according to the actual update frequencies of target Web applications.

3.2. Interface Configuration

Web services are self-contained and self-describing, and communicate using open protocols like HTTP. The most-used style architectures of Web services are SOAP and REST. SOAP stands for *Simple Object Access Protocol*. Google implements their Web services to use SOAP, and we can find SOAP Web services in a number of enterprise software. REST stands for *Representational State Transfer*. A number of new web services are implemented using a REST style architecture these days rather than a SOAP one, such as the Web services of Yahoo, Flickr and del.icio.us. Compared with SOAP, REST is lightweight and easy to build, and provides the readable results.

In the proxy server, we use the standard RESTful Web service interfaces between the users and the constructed Web services. The users can use the Web services by sending the proper parameter values to the Web services and get the response data through them.

For the interfaces between the constructed Web services and the target Web applications, if the target Web applications use dynamic URLs, we get the requests from the

pages with similar layout no matter what keywords we enter. We can reuse the designed extraction pattern of a typical Web page to extract the partial information from these similar response Web pages. We developed a Web application for creating extraction pattern quickly and easily by GUI. We select the target part and the data type by mouse instead of parsing the HTML document of Web page to find out the paths manually.

1. We get a typical response Web page, and define the names and data types for each required part: news title part is *NewsTitle* of continuous text type, news link part is *NewsLink* of continuous link type, and publication date part is *PublicationDate* of continuous text type. We use the following format to save the names, data types and paths of the selected parts as the extraction pattern.

```
<NewsTitle type="continuous text">BODY:0/DI
V:1/DIV:0/DIV:0/DIV:2/DIV:0/A:0/</NewsTitle>
<NewsLink type="continuous link">BODY:0/DIV
:1/DIV:0/DIV:0/DIV:2/DIV:0/A:0/</NewsLink>
<PublicationDate type="continuous text">BODY:0/DIV:
1/DIV:0/DIV:0/DIV:2/DIV:0/SPAN:1/</PublicationDate>
```



Figure 9. Target parts selection and extraction pattern generation

2. We configure the Web service interface, and give the parameter *query*.
3. The Web service receives the request URL from the user and emulate the submit to receive the response Web page.

`http://VirtualWebServiceProxyServer/CNN/search?query=Tokyo`

4. The Web service extracts the partial information from the response Web page and returns the user the news about *Tokyo* in the first result page as shown in Fig. 10.

We can use the similar method to construct a Web service for ACM Calendar of Events [7]. The users can search for the events by date or text keyword as shown in Fig. 11.

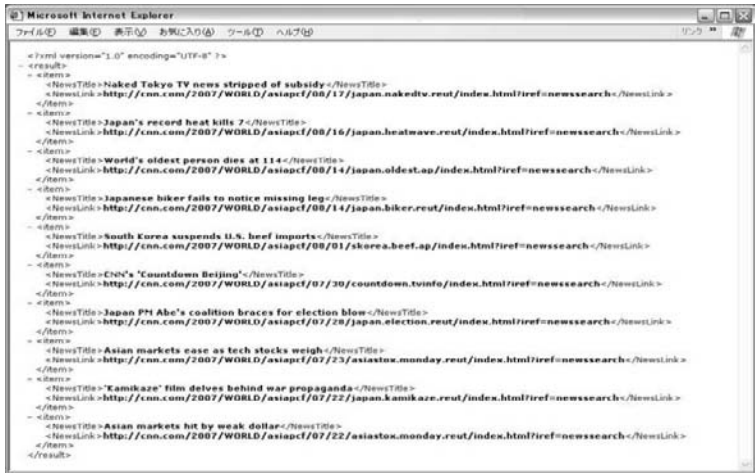


Figure 10. A response of CNN news search from the Web service



Figure 11. ACM Calendar of Events and search result from constructed Web service

For the static URL type, we construct a Web service for querying information from BBC Country Profiles based on the static extraction. Compared with the dynamic extraction, we collect the URLs of target Web pages from the drop-down list and extract the selected parts irregularly to update the information about the new selected country leaders. Fig. 12 shows the integration of constructed Web services with Google Maps.

Our approach is applicable to the general Web applications and provides an extraction pattern creation tool with GUI. The extraction range includes text, link, picture and other multimedia files of Web pages with static URLs or dynamic URLs, and our algorithm is applicable to the extraction from lists or tables in Web pages. We set the extraction frequencies for Web services and decrease the load of our proxy server. Although we realize the quick construction of Web services, our approach still depends on some manual work and is not robust enough when the Web applications change the layout of Web pages. During the interface configuration, it still needs some manual operations. Besides, the Web applications are designed for browsing by users, not for the parsing by computer program. It is difficult for us to select the desired parts from the Web pages of

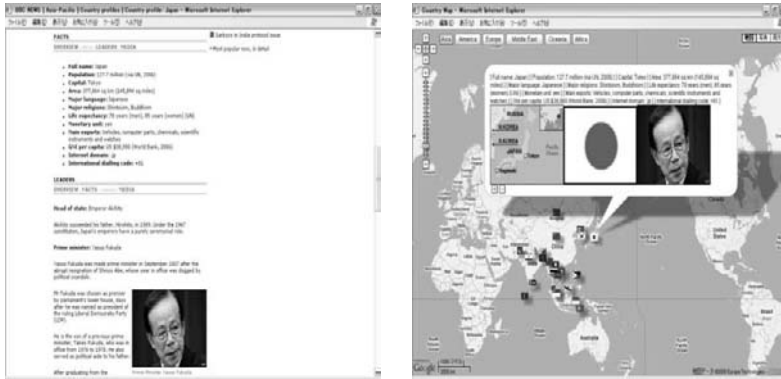


Figure 12. The original Web page and the integration of constructed Web service with Google Maps

some Web sites because the desired parts and undesired parts are intermingled with each other and they have the similar paths.

5. Related Work

Taking advantage of the fact that a great and increasing number of Web sites have their pages automatically generated from database, a number of approaches have been proposed to analyze the structure of the Web pages of these Web sites with the purpose of manual or semi-automatic example-based information extraction and Web service construction ultimately.

To the information extraction, XSLT [8] uses the defined path patterns to find nodes that match with given paths repeatedly and outputs data by using information of the nodes and values of variables. Similarly, ANDES [9] is a XML-based methodology to use the manually created XSLT processors to realize the data extraction. PSO [10] is an approach to extract the parts of Web pages. It keeps the view information of the extracted parts by using the designated paths of tree structures of HTML documents. These Web knowledge extraction systems need the users to find out the paths of the desired parts from the HTML document by hands. IEPAD [11] proposes an automatic pattern discovery system. The users select the target pattern that contains desired information from the discovered candidate patterns, then the extractor receives the target pattern and a Web page as input, and applies pattern-matching algorithm to recognize and extract the information. [12] proposes an approach that provides the users a GUI to select the desired parts: text or image, uses the designated paths to extract the partial information from the similar Web pages, and finally returns the users a XML format result. However, this system does not support the link information extraction, which is important for the extraction of the information like the news search result of CNN.

To the Web service construction, HTML2RSS [13] is a Web service to automatically generate RSS feeds from HTML documents that consist of time-series items such as blog, BBS, chats and mailing lists. However, it is limited to the Web pages that consist of list of data items with similar data structures or special data structures. Toshiba Web Service Gateway [14] parses the response HTML page by flexible tree-style and returns the extracted data. It allows the users to try all the HTML parsers available one by one

in the Web Service gateway to find which is the most suitable for parsing the response HTML pages. However, it is difficult for the users to develop the personalized parsers because the users need to find out the tags representing the desired parts one by one. GridXSLT [15] is an implementation of the XSLT language designed for large-scale parallel processing in grid computing environments. It provides a simplified approach to Web service development by implementation of XSLT documents, which need the programming of users. Pollock [16] can create a virtual Web service from FORM-based query interface of Web sites. It generates wrappers using XWrap, and WSDL file using Web site-related information, then publishes the details of the virtual Web service into UDDI, but this system needs the users to parse the HTML documents of the FORM-based Web pages.

Compared to these developed work, our approach realizes the extraction of almost all kinds of partial information such as text, object and link, and provides a GUI for easy part selection and data type definition. We can construct a Web service quickly and easily, and all the process of Web service construction does not need too much manual work and programming.

6. Conclusion and Future Work

Our approach is based on the partial information extraction, resulting table query and interface configuration. We transform the target Web page into the XML-format resulting table and respond to the requests of users with the field values. Our approach is applicable to the general Web applications including the information of link and multimedia files in table structure.

As future work, we will modify our approach to extend the types of applicable Web applications and the processes of similar paths reuse to intensify the extraction when Web sites change the layout of Web pages.

Finally, we will construct many types of Web services to develop the information integration systems by combination of the Web services.

References

- [1] CNN. <http://www.cnn.com>.
- [2] BBC Country Profiles. http://news.bbc.co.uk/2/hi/country_profiles/default.stm.
- [3] Yahoo Finance. http://biz.yahoo.com/ic/ind_index.html.
- [4] HtmlUnit. <http://htmlunit.sourceforge.net/>.
- [5] Rakuten. <http://www.rakuten.co.jp/>.
- [6] BBC News Click 2007 video archive. http://news.bbc.co.uk/2/hi/programmes/click_online/6262703.stm.
- [7] ACM Calendar of Events. <http://campus.acm.org/calendar/>.
- [8] Michael Kay. *XSL Transformations Version 2.0*, 2007. <http://www.w3.org/TR/xslt20/>.
- [9] Jussi Myllymaki. Effective Web data extraction with standard XML technologies. In *The Proceedings of the 10th International Conference on World Wide Web*, 2001.
- [10] Tetsuya Suzuki and Takehiro Tokuda. Path set operations for clipping of parts of Web pages and information extraction from Web pages. In *The 15th International Conference on Software Engineering and Knowledge Engineering*, 2003.
- [11] Chia-Hui Chang and Shao-Chen Lui. IEPAD: Web information extraction based on pattern discovery. In *The Proceedings of the 10th International Conference on World Wide Web*, 2001.
- [12] Hao Han and Takehiro Tokuda. A personal Web information/knowledge retrieval system. In *The 17th European-Japanese Conference on Information Modeling and Knowledge Bases*, 2007.

- [13] Tomoyuki Nanno and Manabu Okumura. HTML2RSS: Automatic generation of RSS feed based on structure analysis of HTML document. In *The Proceedings of the 15th International Conference on World Wide Web*, 2006.
- [14] Hoang Pham Huy, Takahiro Kawamura, and Tetsuo Hasegawa. How to make Web sites talk together - Web service solution. In *The Proceedings of the 14th International Conference on World Wide Web*, 2005.
- [15] Peter M. Kelly, Paul D. Coddington, and Andrew L. Wendelborn. A simplified approach to Web service development. In *Proceedings of the 2006 Australasian workshops on Grid computing and e-research*, 2006.
- [16] Yi-Hsuan Lu, Yoojin Hong, Jinesh Varia, and Dongwon Lee. Pollock: Automatic generation of virtual Web services from Web sites. In *the Proceedings of the 2005 ACM symposium on Applied computing*, 2005.

A News Index System for Global Comparisons of Many Major Topics on the Earth

Tomoya NORO, Bin LIU, Yosuke NAKAGAWA, Hao HAN and Takehiro TOKUDA

Department of Computer Science, Tokyo Institute of Technology, Japan

Abstract. In this paper, we propose a news index system which supports users who would like to observe difference in various topics (e.g. politics, economy, education, and culture) among countries/regions. General news sites just provide news articles and we can only read articles which we are interested in by using a keyword search engine or selecting some topic categories provided by each site. Our system has a large index word list, a news index database, and a news directory system. The word list is constructed, expanded, and updated by collecting topic keywords from various Web sites. The news directory system consists of the word list. News articles are collected by using keyword search engines which news sites provide, then index of the collected articles is stored in the database and classified by the news directory system. We can see the difference in various topics among countries/regions by observing co-occurrence of two or more words in the word list.

Keywords. News index system, index word list construction, news article page collection, news article extraction, news directory

Introduction

We can read a lot of news articles on the Web provided by various news sites. Since these articles cover various topics on the earth, we can see difference in a topic among countries/regions by collecting and classifying news articles related to the topic and the countries/regions. For example, if we collect news articles about whaling and classify them by country/region, we can find countries/regions interested in the topic and see difference among countries/regions related to the position on the topic. However, it is a time-consuming task. Our goal is to construct a news index system for supporting users who would like to see such difference.

When we would like to read some articles we are interested in, we usually search for articles by using a keyword search engine or selecting some topic categories provided by each news site. Classifying articles into some topics is useful for those who cannot come up with appropriate keywords to find intended articles. However, it has some problems.

1. Each site provides its own topic word (topic category) list and articles are classified according to the list in its own way. We cannot search for the intended articles in the same way regardless of news sites.

2. The word lists are not always up to date and do not cover all of the topics in the world since such lists are usually constructed and maintained manually. For example, it is difficult to update a list of country/region leaders' names immediately following a leader change since it occurs frequently (i.e. it will take place somewhere in a few days on average).

When we construct an index of news articles on the Web, we encounter another problem. This process consists of the following parts.

1. Crawl Web pages in news sites.
2. Determine if the obtained Web pages include news articles.
3. Extract a news article body from each of the news article pages.
4. Classify the articles into some categories.

The first process needs to keep running regularly to collect all of the latest news articles since they will soon be deleted or moved to other places which we cannot reach easily from the top page. However, crawling Web pages is a time-consuming task, and, if we would like to crawl a large number of news sites, it is difficult to keep running the process since it sometimes fails due to network failure and server down. The second process is also time-consuming since there are many pages which do not include any news articles, such as pages for advertisement, video, and weather forecast. Additionally, news article body extraction (the third process) is not an easy task. In general, news article body is extracted from a news article page by a template-based method, i.e. structure of some news article pages in each news site are analyzed and its template is created, then news article body is extracted by using the template. In this method, the template needs to be updated if any changes in news article page structure has been made.

In order to solve the first problem (with topic word lists), we propose an automatic creation, expansion, and update of a topic word list for all news sites. We construct and expand the word list automatically by collecting words from some Web sites, then update it by watching the sites whether any changes are made. A directory for news article classification is constructed from the word list.

For the second problem (with news article index construction), we present a new approach for constructing news article index quickly. We utilize keyword search engines which news sites provide. Instead of crawling the news site, each word in the topic word list is given to the search engine and news article pages which include the word are collected. Since we can get only news article pages, the second step of the process of constructing news article index will be carried out faster. Since our method for extraction of news article body does not require any news article page structure templates, we do not have to analyze news article page structure of each site in advance, and we do not have to check if any changes in news article page structure has been made either.

After that, the obtained articles are classified by the news directory mentioned above. We implemented and evaluated a news index system, which supports users who would like to observe difference in various topics (e.g. politics, economy, education, and culture) among countries/regions using word co-occurrence.

The organization of the rest of this paper is as follows. First of all, an overview of the news index system is presented in Section 1, then details of each part of the system is described in Section 2, 3, and 4. Implementation and evaluation of the system is shown in Section 5. Finally, we conclude this paper and give some directions to future work in Section 6.

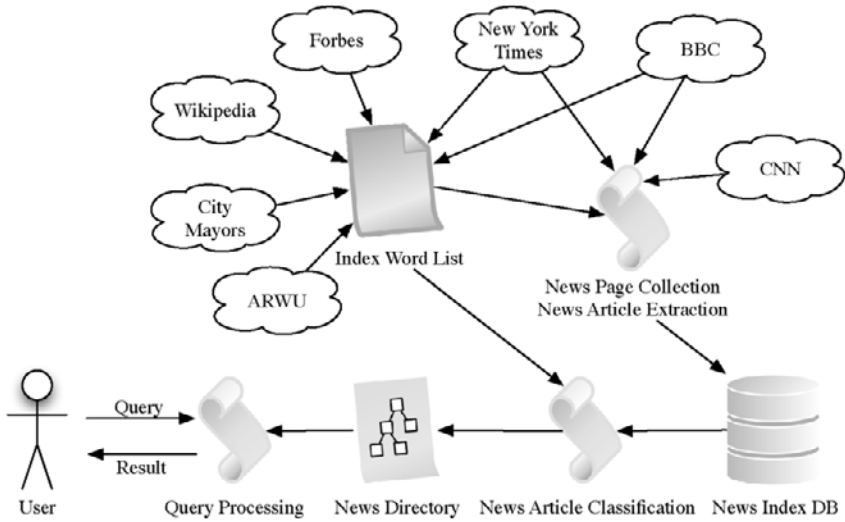


Figure 1. An overview of the news index system.

1. System Overview

The news index system consists of the following four parts (Fig. 1).

1. Construction, expansion, and update of an index word list
We construct an index word list based on a topic word list provided by a news site. Since the list is not sufficient, we automatically get some words from other Web sites and add them to the list. The Web sites are always watched, and the list is updated if any changes are made in the sites.
2. News article page collection and news article extraction
Each word in the index word list is given to a keyword search engine provided by a news site, and obtain search result pages. After URL of each news page is extracted, we get the page, extract the title and the article body, and store the index information to an news index database.
3. News article classification
The news articles are classified by a news directory system. The directory system consists of the index word list. If the list is updated, the directory system is also updated and the classification process is carried out again.
4. Query processing
Users can search for news articles they are interested in by following the news directory system or by giving keywords directly.

2. Index Word List

The New York Times (NYT) provides the Times Index, and news articles published by NYT are classified by subject, place, organization, and personal name (the Times

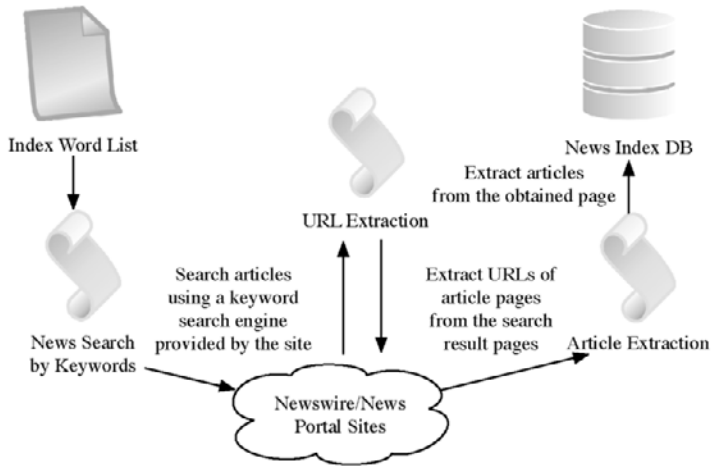


Figure 2. An overview of the news article collection.

Topics) [1]. Although the index is useful for searching for articles published by NYT, as mentioned previously, it has problems with cross-site indexing and real-time update.

In our news index system, beginning with the Times Topics, we get some words about names of countries/regions, capital cities, other major cities, leaders of countries/regions, celebrities, companies, universities, international organizations, economic indicators, crimes, etc from some external Web sites such as Wikipedia [2] (the details are described in Section 5) and add them to the list. The external Web sites are always watched and, if any changes are made in the sites, the list is updated.

3. News Article Collection Through News Search Engines

This process consists of the following parts (Fig. 2).

1. News article search by keywords.
2. Extraction of URL, title, and publication date of each news article.
3. Extraction of body of the news article.

The details of these processes are described in the rest of this section.

3.1. News Article Search by Keywords

Instead of crawling news sites, we utilize a keyword search engines provided by each site. Queries are sent to each search engine by using words in the index word list. In general news sites, we can get search result pages via GET method (not only POST method). For example, we can get search result pages from CNN by sending the following request URL, which indicates the keyword is “Olympic” (“query = Olympic”), search for news articles in the international edition is requested (“type = news” and “intl = true”), the results are sorted by date (“sortBy = date”), and the second search result page is requested (“currentPage = 2”).

<http://search.cnn.com/search?query=Olympic&type=news&sortBy=date&intl=true&nt=null¤tPage=2>

This process generates a request URL for each news site including some parameters (e.g. keywords, sort order) and sends it to the search engine.

3.2. Extraction of URL, Title, and Publication Date of Each News Article

After search result pages are obtained, URL of each news article page is extracted. Han et al. proposed a tree structure based method for extraction of partial information from Web pages with similar structure [3]. Since all of the search result pages have the similar structure, we can apply the method to this process. Once one URL of news article page in a search result page is selected and the path from the root node (i.e. “body”) is obtained, all news article URLs can be obtained from search result pages produced by the same news site in the following way.

1. Find a node N corresponding to the path in the search result page.
2. Let P and S be the parent node of N and the subtree of P respectively.
3. Let L be a set of nodes whose paths include the path of S .
4. If $|L| \geq 2$ or P is “body”, return L . Otherwise, let P and S be the parent node of the old P and the subtree of the new P respectively, then go to step 3.

Each node in the set L corresponds to a node of news article page URL in the search result page. News titles and publication dates in the search result page are also extracted in the same way.

Since the number of search results is large and the display usually extends to multiple pages, the process described above will be repeated several times changing the search result page number (in the case of CNN, the value of the parameter “currentPage”) until one of the following condition is satisfied.

1. The search result page does not exist (the Web server replies “page not found”).
2. No URL and title of news article can be extracted.
3. The URLs and titles extracted from the search result page are the same as those of the previous page.

3.3. Extraction of News Article

After a news title and URL of each article page are obtained, this process gets the page and extracts body of the article. In our method, we do not have to analyze news article page structure of each news site in advance, and we do not have to check if any changes in news article page structure has been made. Our method is robust to any changes in news article page structure.

The phase of the news article extraction consists of the following two parts.

1. Detection of news titles

The process detects position of a news title in the obtained article page. It is helpful for the following process since body of a news article is usually preceded by its title. Note that the title extracted in this process is used only for the next process (news article body extraction). The news title shown in the search result page is stored in our news index database.

Since the title shown in the search result page is not always the same as the real title in the news article page, exact match is not appropriate for this process. Instead, for each node n in the news article page (an HTML document), we calculate similarity score described below (t is the news title shown in the search result page).

$$\text{Sim}(n, t) = \frac{(\text{key}(n, t))^2}{\text{word}(n) \times \text{keysize}(t)}$$

$\text{word}(n)$ and $\text{key}(n, t)$ are defined as follows.

- (a) If n is a leaf node, $\text{word}(n)$ and $\text{key}(n, t)$ are the number of words and “key-words” covered by the node n respectively (“keywords” are words in the news title t).
- (b) If n is not a leaf node,

$$\text{word}(n) = \sum_{n' \in \text{Child}(n)} \text{word}(n'), \quad \text{key}(n, t) = \sum_{n' \in \text{Child}(n)} \text{key}(n', t)$$

$\text{keysize}(t)$ indicates the number of words in the news title t (i.e. the size of the “keyword” set).

If the score is higher than a predetermined threshold, the string covered by the node n is judged as a news title. If there is no node whose score is higher than the threshold, no string is judged as a news title. On the other hand, if there are more than one node with higher score than the threshold, all of the strings covered by the nodes are judged as news titles.

2. Extraction of body of the news article

The process detects a part of the news article body and extract the whole body. Since body of a news article is usually preceded by its title, the process tries to find the news article body in some “contents ranges” at first, and, if it cannot find out the body in the range, it tries to find the body in a “reserve range”. “Contents range” and “reserve range” are parts which might include the news article body. They are determined as follows.

- If only one string is judged as a news title in the previous process, the following part and the preceding part are a contents range and a reserve range respectively (Fig. 3(a)).
- If no string is judged as a news title, the whole part of the news article page is a contents range and no reserve range exists (Fig. 3(b)).
- If more than one string are judged as news titles, for each of the strings except the last string, range of between itself and the next string is a contents range. The part preceded by the last string is also a contents range. The part followed by the first string is a reserve range (Fig. 3(c)).

Firstly, we specify a part of news article body. For each leaf node with non-link text n in each of the contents ranges, we calculate possibility score described below.

$$\text{Pos}(n) = \sum_{n' \in B(n)} \text{word}(n') \times \left(\sum_{n' \in B(n)} \text{key}(n', t) + 1 \right)$$

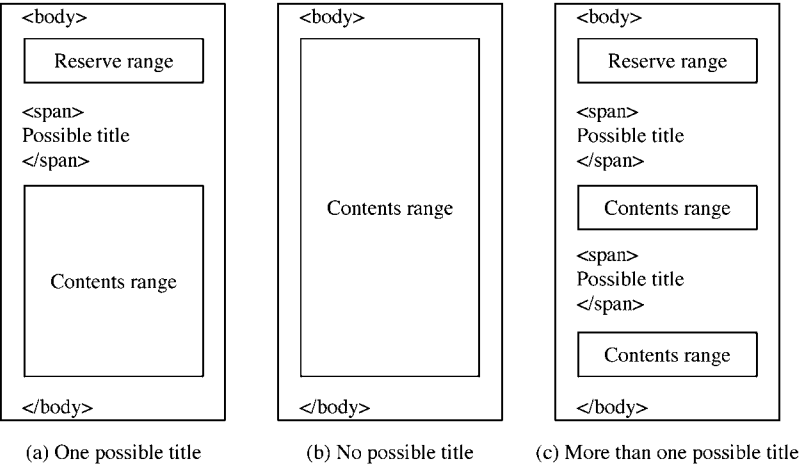


Figure 3. Contents range and reserve range.

where $B(n)$ indicates a set of nodes defined as follows.

- (a) The node n is in $B(n)$.
- (b) If a leaf node n' satisfies all of the following conditions, the node n' is also in $B(n)$.
 - i. The node n' is in the same range as the text node n .
 - ii. The node n and n' are siblings, or their parents' nodes are siblings.
 - iii. The node n' or its parent node is one of the following nodes: #text, "span", "a", "p", "ul", "ol", "dd", "dt", "strong", "h1", "h2", "h3", "h4", "b".

If there are one or more than one nodes with higher score than a predetermined threshold, the string covered by the node with the highest score is judged as a node which covers a part of the news article body. If there is no node with higher score than the threshold, we calculate the score for each leaf node with non-link text in the reserve range and the string covered by the node with the highest score among all of the nodes in the contents ranges and the reserve range is judged as a node which covers a part of the news article body.

After a node which covers a part of news article body is specified, the whole article body is extracted. Since a news article body is usually a continuous text, it can be extracted by taking leaf nodes around the specified node. However, in some cases, some information which is not related to the article, such as advertisement, is inserted in the article body. In order to avoid taking such information, only leaf nodes around the specified node which satisfies all of the following conditions are taken.

- (a) The target node and the specified node are siblings, or their parents' nodes are siblings.
- (b) The target node or its parent node is one of the following nodes: #text, "span", "a", "p", "ul", "ol", "dd", "dt", "strong", "h1", "h2", "h3", "h4", "b".

Finally, we get a list of nodes which cover the whole news article body. The whole body can be extracted by getting the node value (i.e. text) from each node in the list.

4. Classification of News Articles by a News Directory System

This process consists of the following two parts.

1. Construction of a news directory system

A news directory system for classification of obtained news article is automatically constructed from the index word list. The directory system has multi-level tree structure, and the number of levels is determined by how many levels the index word list is classified into. For example, if the list has several categories, such as place names, personal names, and company/organization names, and every word in the list is assigned to one of the categories, the number of levels of the news directory system will be two (Fig. 4). If the list does not have any categories and all words are just put into the list, a directory system with one-level flat structure will be constructed.

Once a news directory system is constructed from the index word list, the process do not have to be carried out again until the word list is updated. The directory system will be re-constructed when any changes in the list are made.

2. Assignment of obtained news articles

The process assigns obtained news articles to corresponding news directories. Liu et al. proposes a quick automaton-based method for it [4]. We apply the method to this process. Definition of each directory is determined by a word/phrase related to the directory: if the directory is produced from a word/phrase W , news articles which includes the word/phrase W are assigned to the directory.

5. Implementation and Evaluation

5.1. Web Sites Which Are Used for Index Word List Expansion

As described in Section 2, the index word list is constructed by expanding the Times Topics provided by the New York Times. We collected words about countries/regions, leaders of countries/regions, companies, crimes, economic indicators, etc from some Web sites other than New York Times. The external Web sites we used for index word list expansion are listed in Table 1. The total number of words is about 17,000.

5.2. Collection and Extraction of News Articles

In order to evaluate on our news article extraction method, we collected news article index from CNN. It has its own database which includes news article published in the past about 10 years. Articles in the databases can be obtained through the keyword search engine they provide. Thresholds for the similarity score $\text{Sim}(n, t)$ and the possibility score $\text{Pos}(n)$ are set as 0.6 and 100 respectively. 96,095 news article pages published in the past 5 years (from January 1, 2003 to December 31, 2007) could be obtained.

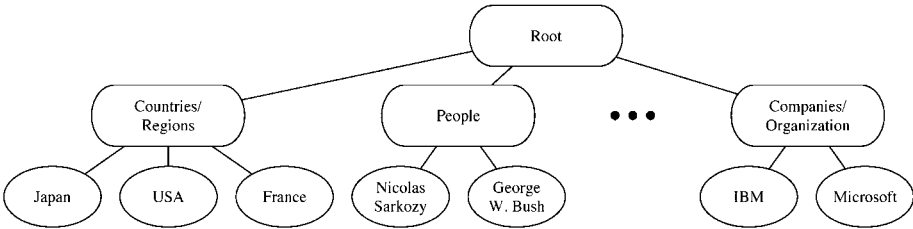


Figure 4. An example of a news directory system with two-level tree structure.

Table 1. List of Web sites which are used for index word list expansion

Web site	Category	# words	Web site	Category	# words
BBC Country Profiles [5]	Countries/regions	243	Wikipedia	Entertainment	104
	Capital cities	207		Music	190
	Leaders of countries/regions	207		Sports	622
	International organizations	173		Game	52
Forbes [6]	Celebrities	100		Hobby	441
	Powerful women	100		Festivals	883
	Companies	2,000		Film awards	160
City Mayors [7]	Major cities	300		Film	51
ARWU [8]	Universities	510		Theatre	35
Wikipedia	States in USA	50		Ethnic groups	953
	Ecology	128		Human attributes	62
	Psychology	105		Human relationships	43
	Sociology	57		Self	169
	Zoology	415		Internal organs	56
	Economics	283		Bony areas	50
	Politics	135		Muscle	483
	Robotics	126		Religion	60
	Computer science	122		Disease	2514
	Computer programming	77		Drugs	400
	Artificial intelligence	60		Crime	119
	Internet	471		Education	48
	Economical indicators	69		Jobs	911
	Finance	45		Energy	15
	Business	83		Landforms	120
	Marketing	229		Life cycle	17
	Food	309		Manufacturing	23
	Fruit	529		Nuclear technology	77
	Nutrition	383		Military	34
	Vegetables	276		Transportation	212
	Agriculture	105		History	59
	Architecture	125		Medieval history	86
	Total				17,100

Table 2. Result of news article page extraction

	Success	Partially-extracted	Non-extracted
# of pages	192	6	2

Table 3. A list of news sites which we collected news article index from

Country/region	Name	URL
United States	CNN	http://www.cnn.com/
	ABC News	http://www.abcnews.go.com/
	New York Times	http://www.nytimes.com/
	Washington Post	http://www.washingtonpost.com/
	Wall Street Journal	http://online.wsj.com/
	Chicago Tribune	http://www.chicagotribune.com/
	USA Today	http://www.usatoday.com/
	United Press International	http://www.upi.com/
	Los Angeles Times	http://www.latimes.com/
	CNet	http://www.cnet.com/
Canada	CNW Group	http://www.newswire.ca/
	CBC	http://www.cbc.ca/
United Kingdom	BBC	http://www.bbc.co.uk/
	Guardian Unlimited	http://www.guardian.co.uk/
France	Euro News	http://www.euronews.net/
	France 24	http://www.france24.com/france24Public/en/
Japan	Mainichi Daily News	http://mdn.mainichi.jp/
China	Xinhuanet	http://news.xinhuanet.com/english/
South Korea	Chosun Ilbo	http://english.chosun.com/
Australia	News.com.au	http://www.news.com.au/
Qatar	Al Jazeera	http://english.aljazeera.net/
Mauritius	All Africa	http://allafrica.com/

Accuracy of news article body extraction is evaluated on 200 randomly selected news article pages.

The result is shown in Table 2. 2 news articles were not extracted since the corresponding pages could not be obtained (the server responded “page not found”). In 6 pages, some parts of the news article were not extracted. All of them included itemization, and a node corresponding to each of the items and a node with the highest possibility score $Pos(n)$ in the page (i.e. a node which is specified as a part of news article body in the first step of the extraction process) were not siblings.

Actually, we have been collecting news articles not only by the collection method described in Section 3 but also by using RSS provided by each news site. Since a title, publication date, and URL of each news article can be easily obtained from RSS, we can extract the news article body from each news article page in the same way as our extraction method.¹ News sites which we have been collecting news articles from are listed in Table 3.

¹Unlike our collection method through search engines, we have to keep running the alternative method regularly since only recent news information is usually included in RSS.

5.3. Construction of a News Directory System

A news directory system is constructed from the index word list. As described in Section 4, basically, definition of each directory is determined by a word/phrase related to the directory. However, some words such as names of people, and countries/regions have some aliases. In order to cover the aliases, we add some aliases of personal and country/region names to corresponding directory definition. For example, definition of a directory “George W. Bush” is “((George W. Bush) OR (President George Bush) OR (President Bush))”.

5.4. Query Processing

Users can search for news articles they are interested in by following the news directory system or by giving keywords directly. Additionally, our system can provide co-occurrence frequency of two or more words by calculating intersection of sets of news articles which are assigned to the corresponding news directories or searched by keywords. For example, we can observe co-occurrence of one or two country/region names and a word related to an interested topic. We can also see monthly/yearly variation of the frequency.

Table 4 and 5 show monthly frequency of country/region names together with one of USA, Russia, UK, France, China and Japan (i.e. permanent members of the U.N. Security Council and Japan) from July to December in 2007. Numbers enclosed in parentheses indicate the number of news articles. **Bold**, underlined, and *italic* country/region names indicate permanent members of the U.N. Security Council, neighboring countries/regions, and disputed countries/regions. It may be natural that frequency of each country/region together with a member of the U.N. Security Council or a neighboring country/region is high. On the other hand, we can see some interesting points. For example, co-occurrence frequency of France and Chad is ranked high in the last 3 months although it had not been ranked in the top-10 before then. Actually, many news articles about “the Zoe’s Ark incident (affair)” (a French aid group, the Zoe’s Ark, was preparing to fly more than 100 Chadian children to France with a view to having them adopted, and the head of the group and others were arrested) are published in this period. Co-occurrence of Japan and Myanmar in October 2007 is also higher than that in other period. In this period, Japan canceled a grant to Myanmar to protest the nation’s crackdown on pro-democracy demonstrators (a Japanese journalist was killed in the incident).

Table 6 and 7 show monthly frequency of country/region names together with a topic word (smoking, H5N1, whale, or kidnap). We can see which countries/regions are strongly related to each topic. For example, countries/regions where H5N1 virus had spread (e.g. Indonesia, Viet Nam, China, and UK) are ranked high. We can guess that the virus spread especially in November and December. In the case of the topic word “whale”, pro/anti-whaling countries/regions (especially Japan and Australia) are listed.² When it comes to the word “kidnap”, although many news articles about kidnapping in Afghanistan were published until September, the frequency decreased after October. On the other hand, the number of articles about kidnapping related to France and Chad increased since October, which reflects occurrence of “the Zoe’s Ark incident” mentioned

²Although we cannot see which countries/regions are for/against whaling only from the result, at least, the countries/regions which appear in many news articles with “whale” would not be sitting on the fence.

Table 4. Co-occurrence of two country/region names

Time period (# articles)	Jul. 2007 (197,649)	Aug. 2007 (201,425)	Sep. 2007 (206,508)	Oct. 2007 (205,576)	Nov. 2007 (196,744)	Dec. 2007 (176,386)
USA	1 <i>Iraq</i> (1,086)	<i>Iraq</i> (1,024)	<i>Iraq</i> (1,584)	<i>Iraq</i> (999)	<i>Iraq</i> (629)	<i>Iraq</i> (499)
	2 UK (421)	China (409)	China (441)	China (374)	China (302)	Iran (319)
	3 China (306)	UK (396)	UK (398)	UK (308)	UK (267)	China (279)
	4 Iran (237)	Iran (271)	Iran (360)	Iran (304)	Pakistan (263)	UK (249)
	Australia (167)	Australia (237)	Australia (310)	Turkey (274)	Canada (247)	Australia (214)
	Canada (165)	<i>Afghanistan</i> (218)	Canada (224)	Russia (181)	Australia (233)	Russia (166)
	Russia (155)	France (202)	North Korea (194)	<i>Afghanistan</i> (171)	Iran (214)	Canada (158)
	Japan (147)	Japan (191)	Russia (190)	Canada (169)	Japan (199)	Pakistan (139)
	Brazil (116)	Canada (161)	Germany (189)	Japan (162)	Israel (145)	<i>Afghanistan</i> (138)
	10 Pakistan (107)	<i>Mexico</i> (158)	<i>Afghanistan</i> (129)	Australia (160)	France (129)	Japan (130)
Russia	1 UK (287)	USA (137)	USA (190)	USA (181)	USA (121)	USA (166)
	2 USA (155)	Georgia (100)	UK (117)	Iran (128)	UK (69)	Iran (108)
	3 China (43)	China (52)	China (115)	UK (124)	Germany (39)	UK (71)
	4 France (30)	UK (52)	Germany (67)	Ukraine (44)	Georgia (39)	France (50)
	5 Serbia (24)	Canada (35)	Israel (57)	China (43)	Israel (38)	Serbia (37)
	6 South Korea (23)	<i>Belarus</i> (26)	France (56)	France (43)	China (35)	China (36)
	7 Iran (22)	Israel (23)	Australia (52)	Estonia (41)	Ukraine (32)	Poland (27)
	8 Germany (19)	Czech Republic (17)	Iran (43)	<i>Kazakhstan</i> (31)	Croatia (21)	Japan (25)
	9 Japan (18)	Iran (15)	Spain (31)	Malaysia (29)	Serbia (18)	Germany (22)
	10 Poland (17)	Japan (14)	Indonesia (26)	Germany (24)	Australia (17)	Czech Republic (15)
UK	1 USA (421)	USA (396)	USA (398)	USA (308)	USA (267)	USA (249)
	2 Australia (341)	India (168)	<i>Iraq</i> (147)	France (255)	Italy (187)	<i>Iraq</i> (164)
	3 India (314)	Germany (155)	France (145)	<i>Iraq</i> (189)	Australia (109)	<i>Sudan</i> (140)
	4 Russia (287)	<i>Iraq</i> (145)	Australia (139)	Australia (176)	Croatia (108)	Sri Lanka (136)
	5 <i>Iraq</i> (181)	Australia (139)	Portugal (122)	Russia (124)	France (101)	<i>Afghanistan</i> (115)
	6 France (162)	<i>Afghanistan</i> (124)	Russia (117)	South Africa (114)	Germany (100)	France (109)
	7 Germany (87)	France (112)	South Africa (106)	Germany (91)	<i>Sudan</i> (92)	Italy (86)
	8 Nigeria (86)	China (67)	Germany (104)	Colombia (85)	Colombia (76)	Australia (80)
	9 <i>Afghanistan</i> (67)	South Africa (63)	China (93)	<i>Ireland</i> (75)	Russia (69)	Russia (71)
	10 China (66)	<i>Ireland</i> (58)	Germany (83)	Estonia (63)	South Africa (66)	Germany (59)

Table 5. Co-occurrence of two country/region names (cont.)

Time period (# articles)	Jul. 2007 (197,649)	Aug. 2007 (201,425)	Sep. 2007 (206,508)	Oct. 2007 (205,576)	Nov. 2007 (196,744)	Dec. 2007 (176,386)
China	1 USA (306)	USA (409)	USA (441)	USA (374)	USA (302)	USA (279)
	2 <u>Japan</u> (66)	<u>Japan</u> (119)	<u>Australia</u> (132)	<u>India</u> (75)	<u>Japan</u> (69)	<u>Japan</u> (112)
	3 UK (66)	UK (67)	Russia (115)	<u>Myanmar</u> (61)	<u>India</u> (65)	<u>India</u> (68)
	4 Russia (43)	<u>Australia</u> (62)	UK (93)	<u>Japan</u> (60)	<u>Australia</u> (61)	UK (50)
	5 <u>India</u> (39)	<u>Germany</u> (58)	<u>Japan</u> (81)	UK (49)	France (55)	<u>Iran</u> (36)
	6 <u>North Korea</u> (38)	Russia (52)	<u>Germany</u> (75)	Russia (43)	<u>Iran</u> (43)	Russia (36)
	7 <u>Pakistan</u> (34)	<u>India</u> (49)	<u>India</u> (64)	<u>North Korea</u> (37)	UK (41)	<u>Australia</u> (29)
	8 <u>Singapore</u> (28)	<u>Canada</u> (32)	<u>North Korea</u> (55)	<u>Germany</u> (29)	Russia (35)	France (28)
	9 <u>Australia</u> (28)	France (28)	France (54)	<u>Iran</u> (27)	<u>Germany</u> (27)	<u>Germany</u> (22)
	10 <u>South Korea</u> (26)	<u>Sudan</u> (24)	<u>Brazil</u> (52)	<u>South Korea</u> (24)	<u>Singapore</u> (26)	<u>Singapore</u> (21)
France	1 UK (162)	USA (202)	UK (145)	UK (255)	USA (129)	<u>Chad</u> (119)
	2 <u>Germany</u> (101)	UK (112)	USA (125)	USA (101)	UK (101)	UK (109)
	3 USA (81)	<u>Iraq</u> (54)	<u>Iran</u> (112)	<u>Germany</u> (75)	<u>Chad</u> (88)	USA (97)
	4 <u>Spain</u> (77)	<u>Italy</u> (51)	<u>Germany</u> (74)	<u>New Zealand</u> (69)	<u>Germany</u> (71)	<u>Italy</u> (69)
	5 <u>Denmark</u> (66)	<u>Germany</u> (48)	<u>Ireland</u> (63)	<u>South Africa</u> (64)	<u>Spain</u> (65)	<u>Spain</u> (60)
	6 <u>Italy</u> (61)	<u>Spain</u> (43)	<u>Argentina</u> (57)	<u>Spain</u> (59)	China (55)	<u>Germany</u> (52)
	7 <u>Australia</u> (55)	<u>Libya</u> (39)	<u>Australia</u> (56)	<u>Chad</u> (52)	<u>Belgium</u> (39)	Russia (50)
	8 <u>Libya</u> (51)	<u>Panama</u> (35)	Russia (56)	<u>Iran</u> (51)	<u>Iran</u> (37)	<u>Colombia</u> (43)
	9 <u>Switzerland</u> (40)	<u>Iran</u> (30)	China (54)	<u>Argentina</u> (47)	<u>Sudan</u> (31)	<u>Libya</u> (41)
	10 <u>Belgium</u> (37)	<u>Australia</u> (30)	<u>Italy</u> (46)	Russia (43)	<u>Italy</u> (28)	<u>Mauritania</u> (39)
Japan	1 USA (147)	USA (191)	USA (116)	USA (162)	USA (199)	USA (130)
	2 <u>Australia</u> (72)	China (119)	China (81)	China (60)	China (69)	China (112)
	3 China (66)	<u>Australia</u> (54)	<u>Australia</u> (75)	<u>Myanmar</u> (44)	<u>Australia</u> (69)	<u>Australia</u> (70)
	4 <u>South Korea</u> (42)	<u>India</u> (30)	UK (37)	<u>Australia</u> (36)	UK (36)	Russia (25)
	5 <u>North Korea</u> (35)	<u>Germany</u> (24)	<u>Afghanistan</u> (28)	UK (34)	<u>North Korea</u> (32)	<u>South Korea</u> (23)
	6 <u>Viet Nam</u> (24)	<u>South Korea</u> (21)	France (26)	France (30)	<u>Afghanistan</u> (29)	UK (15)
	7 UK (18)	UK (21)	<u>Canada</u> (25)	<u>South Korea</u> (27)	<u>South Korea</u> (26)	<u>North Korea</u> (14)
	8 Russia (18)	<u>Canada</u> (18)	<u>South Korea</u> (23)	<u>North Korea</u> (26)	<u>India</u> (17)	<u>Somalia</u> (13)
	9 <u>Qatar</u> (15)	<u>Mongolia</u> (17)	<u>Myanmar</u> (20)	<u>Afghanistan</u> (23)	<u>Kenya</u> (14)	<u>Iran</u> (8)
	10 <u>Singapore</u> (9)	France (16)	<u>North Korea</u> (19)	<u>Iran</u> (17)	<u>Canada</u> (13)	<u>India</u> (8)

Table 6. Co-occurrence of a topic word and a country/region name

Time period (# articles)	Jul. 2007 (197,649)	Aug. 2007 (201,425)	Sep. 2007 (206,508)	Oct. 2007 (205,576)	Nov. 2007 (196,744)	Dec. 2007 (176,386)
Smoking	1 UK (87)	UK (44)	USA (54)	UK (44)	USA (38)	USA (27)
	2 USA (37)	USA (31)	UK (41)	USA (33)	UK (36)	USA (22)
	3 Kenya (20)	<i>Iraq</i> (13)	<i>Iraq</i> (32)	Nigeria (12)	Australia (21)	France (12)
	4 Australia (14)	Japan (11)	Thailand (14)	Uganda (11)	Nigeria (19)	Australia (8)
	5 Uganda (10)	China (11)	Iran (11)	China (10)	China (10)	Tanzania (7)
	6 Canada (9)	Kenya (9)	Uganda (8)	<i>Iraq</i> (9)	South Africa (10)	Nigeria (7)
	7 Nigeria (8)	Botswana (8)	Canada (7)	Kenya (9)	Italy (9)	Kenya (6)
	8 Germany (8)	Australia (8)	Australia (6)	Australia (9)	Georgia (8)	Canada (5)
	9 Ghana (7)	Uganda (8)	Nigeria (5)	South Africa (7)	Canada (7)	Russia (5)
	10 New Zealand (6)	Germany (6)	Japan (4)	Canada (5)	France (5)	South Africa (5)
H5N1	1 France (8)	Indonesia (10)	China (6)	Indonesia (6)	UK (28)	Indonesia (16)
	2 Germany (7)	Germany (9)	Singapore (5)	Viet Nam (4)	Viet Nam (5)	Poland (16)
	3 Indonesia (5)	Viet Nam (6)	Indonesia (4)	Uganda (2)	Indonesia (4)	Pakistan (15)
	4 Viet Nam (5)	USA (5)	Germany (1)	Canada (2)	Turkey (2)	China (15)
	5 India (4)	Switzerland (4)	Canada (1)	China (1)	China (1)	Myanmar (9)
	6 Turkey (4)	France (3)	Thailand (1)	USA (1)	Myanmar (1)	Viet Nam (8)
	7 USA (4)	UK (3)	Kenya (1)		Japan (1)	Egypt (3)
	8 Myanmar (3)	South Africa (1)				Turkey (2)
	9 South Africa (2)	China (1)				Russia (2)
	10 Thailand (2)	Israel (1)				Germany (2)
Whale	1 UK (13)	UK (14)	USA (7)	Australia (20)	Japan (62)	Japan (96)
	2 USA (10)	USA (6)	UK (4)	Canada (7)	Australia (21)	Australia (60)
	3 Australia (8)	China (6)	Canada (4)	South Africa (5)	Brazil (13)	USA (22)
	4 Japan (6)	Japan (3)	Japan (3)	Georgia (4)	UK (9)	Canada (6)
	5 Russia (5)	Colombia (3)	Australia (2)	UK (4)	USA (8)	UK (4)
	6 South Africa (4)	Australia (2)	South Africa (2)	Japan (3)	Chile (8)	New Zealand (4)
	7 Botswana (3)	Canada (2)	Namibia (1)	Pakistan (2)	South Africa (8)	South Africa (3)
	8 China (2)	India (2)	Uganda (1)	Kenya (2)	Canada (6)	Czech Republic (2)
	9 Namibia (2)	New Zealand (2)	Nigeria (1)	New Zealand (1)	New Zealand (6)	Nigeria (1)
	10 Canada (1)	Kenya (2)	Colombia (1)		Zimbabwe (4)	

Table 7. Co-occurrence of a topic word and a country/region name (cont.)

Time period (# articles)	Jul. 2007 (197,649)	Aug. 2007 (201,425)	Sep. 2007 (206,508)	Oct. 2007 (205,576)	Nov. 2007 (196,744)	Dec. 2007 (176,386)
Kidnap	1 <i>Afghanistan</i> (296)	<i>Afghanistan</i> (277)	<i>Afghanistan</i> (141)	<i>Iraq</i> (120)	USA (113)	France (176)
	2 <i>Nigeria</i> (219)	<i>Korea</i> (207)	USA (122)	USA (112)	<i>Chad</i> (85)	<i>Colombia</i> (123)
	3 USA (206)	USA (149)	<i>Nigeria</i> (90)	<i>Nigeria</i> (103)	France (79)	<i>Chad</i> (101)
	4 <i>Iraq</i> (201)	<i>Nigeria</i> (125)	<i>Iraq</i> (82)	<i>Afghanistan</i> (52)	<i>Iraq</i> (70)	<i>Iraq</i> (98)
	5 <i>South Korea</i> (198)	<i>Iraq</i> (116)	<i>South Korea</i> (46)	<i>Chad</i> (42)	<i>Mexico</i> (60)	<i>Nigeria</i> (72)
	6 <i>Pakistan</i> (184)	<i>Germany</i> (71)	<i>Italy</i> (31)	France (41)	<i>Nigeria</i> (46)	USA (71)
	7 UK (136)	<i>Pakistan</i> (45)	<i>Colombia</i> (27)	UK (35)	<i>Spain</i> (34)	<i>Venezuela</i> (62)
	8 <i>Germany</i> (76)	UK (43)	UK (27)	<i>Pakistan</i> (36)	UK (33)	UK (53)
	9 <i>Philippine</i> (32)	<i>Portugal</i> (22)	<i>Pakistan</i> (26)	<i>Iran</i> (28)	<i>Colombia</i> (30)	<i>Somalia</i> (49)
	10 <i>Italy</i> (32)	<i>Iran</i> (22)	<i>Germany</i> (25)	<i>Sudan</i> (25)	<i>Sudan</i> (23)	<i>Peru</i> (35)

previously. Figure 5 shows total co-occurrence of a topic word and a country/region for each area (i.e. Europe, North America, South America, Asia, Oceania, Africa). We can easily catch the situation mentioned above from the result.

Table 8 shows monthly frequency of country/region name pairs together with a topic word (whale or kidnap). We can see which two countries/regions are involved together in the topic. For example, Japan and some other pro-whaling countries/regions have an argument over whaling with Australia and some other anti-whaling countries/regions, especially in the end of 2007. The result reflects the fact: the number of news articles including the topic word “whale”, and the two countries (i.e. Japan and Australia) increased in November and December. Additionally, we can find that some other countries/regions, such as USA and New Zealand, may be also involved in the argument. In the case of the topic word “kidnap”, we can guess that kidnapping often occurred in Afghanistan and Iraq, and the hostages were Koreans, Americans, Germans, etc. We can also see that the number of news articles with “kidnap”, France and Chad increased since October as we could also see the similar result in the previous two experiments. Additionally, Colombia and Venezuela often appeared together in the topic in November and December. Actually, the president of Venezuela was negotiating with a Colombian rebel group over hostages’ release.

6. Conclusion

We presented a news index system for supporting users who would like to observe difference in various topics among countries/regions using word co-occurrence. The system has the following features.

- Our index word list is constructed and expanded by picking up topic words from various Web sites. Although general news sites have such word lists, they are usually maintained manually. Our index word list is updated automatically if any changes are made in the Web sites.

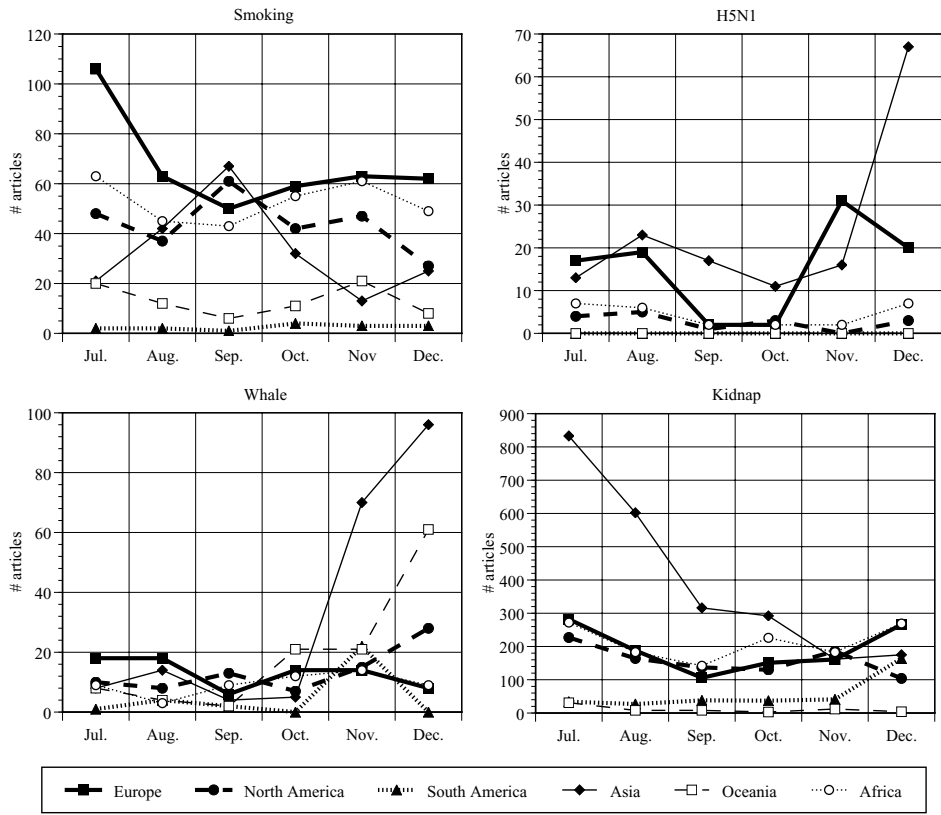


Figure 5. Total co-occurrence of a topic word and a country/region name for each area.

- The system collects news articles from some news sites through keyword search engines provided by the sites. Crawling many news sites regularly is a time-consuming task. Our method does not have to be carried out so often since we can obtain news articles published many months/years ago.
- Our method for extraction of news article body is robust to changes in news article page structure. In general, news article page structure of each news site is analyzed and a template for the site is created in advance, then news article body is extracted by using the template. However, our method does not require any templates, and we do not have to check if any changes in news article page structure has been made.
- The system allows us to know the relationship among countries/regions, people, companies/organizations, etc by retrieving word co-occurrence. General news sites only classify their own news articles by using their own topic list and we can only read news articles classified into each topics.

In the future, we are planning to expand our index word list. We would like to add words/phrases about abstract topics as well as concrete topics.

Currently, definition of each news directory is not determined full-automatically. In the case of countries/regions, people, etc, some aliases are added manually to each definition. We need to consider automatic expansion of news directory's definition.

Table 8. Co-occurrence of a topic word and two country/region names

Time period (# articles)	Jul. 2007 (197,649)	Aug. 2007 (201,425)	Sep. 2007 (206,508)	Oct. 2007 (205,576)	Nov. 2007 (196,744)	Dec. 2007 (176,386)
Whale	1 UK Russia (4)	UK Colombia (3)	Japan Australia (1)	Japan Australia (2)	Australia <u>Australia</u> New Zealand (6)	Japan Australia (45)
	2 Russia Namibia (1)	UK China (2)	UK Colombia (1)	Australia Pakistan (2)	<u>Australia</u> New Zealand (6)	USA Japan (7)
	3 USA South Africa (1)	UK Canada (2)	USA <u>Canada</u> (1)	Australia UK (2)	New Zealand Japan (6)	USA Australia (6)
	4 USA UK (1)	Canada Colombia (2)		UK Pakistan (2)	USA Japan (2)	<u>Australia</u> New Zealand (3)
	5 USA <u>Canada</u> (1)				Australia UK (1)	Australia Canada (2)
	6				UK South Africa (1)	New Zealand Japan (1)
	7				UK Japan (1)	Japan Canada (1)
Kidnap	1 <i>Afghanistan</i> South Korea (178)	<i>Afghanistan</i> South Korea (169)	USA <i>Iraq</i> (45)	USA <i>Iraq</i> (48)	France Chad (52)	France Chad (94)
	2 USA <i>Iraq</i> (97)	Germany <i>Afghanistan</i> (57)	South Korea <i>Afghanistan</i> (32)	France Chad (27)	Colombia <u>Venezuela</u> (46)	Colombia <u>Venezuela</u> (61)
	3 UK Nigeria (75)	USA <i>Iraq</i> (40)	USA <i>Afghanistan</i> (15)	<i>Afghanistan</i> Germany (11)	USA <i>Iraq</i> (31)	USA <i>Iraq</i> (33)
	4 <i>Afghanistan</i> Germany (55)	USA <i>Afghanistan</i> (36)	Pakistan <i>Afghanistan</i> (9)	France <i>Sudan</i> (10)	France Chad (20)	France Somalia (31)
	5 Niger <u>Nigeria</u> (29)	USA South Korea (17)	Niger <u>Nigeria</u> (7)	<u>Sudan</u> <u>Chad</u> (9)	<u>Sudan</u> <u>Chad</u> (20)	France Colombia (26)
	6 USA <i>Afghanistan</i> (24)	Pakistan <i>Afghanistan</i> (14)	<i>Iraq</i> <i>Afghanistan</i> (4)	Niger <u>Nigeria</u> (9)	France <i>Sudan</i> (19)	<i>Sudan</i> <u>Chad</u> (24)
	7 USA Nigeria (18)	Niger <u>Nigeria</u> (12)	Korea USA (3)	Chad Spain (6)	USA Colombia (12)	USA <i>Iraq</i> (21)
	8 Italy <i>Afghanistan</i> (15)	UK Nigeria (7)	South Korea Germany (3)		<i>Iraq</i> <u>Turkey</u> (11)	USA Colombia (16)
	9 Italy South Korea (15)	Nigeria Pakistan (5)			USA Venezuela (8)	France Venezuela (13)
	10 Italy Philippine (14)					USA Venezuela (12)

About half of the news sites which we collected news articles from is located in the US and UK. As a result, the number of news articles related to the two countries tends to be higher than the others. In order to solve the problem, we need to expand the news site list equally. Otherwise, we need to count relative frequency.

References

- [1] New York Times Topics. <http://topics.nytimes.com/>.
- [2] Wikipedia. <http://en.wikipedia.org/>.
- [3] Hao Han, Yohei Kotake, and Takehiro Tokuda. An efficient method for quick construction of web services. In *The 18th European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 183–196, 2008.
- [4] Bin Liu, Pham Van Hai, Tomoya Noro, and Takehiro Tokuda. Towards automatic construction of news directory systems. In *The 17th European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 211–220, 2007.
- [5] BBC Country Profiles. http://news.bbc.co.uk/2/hi/country_profiles/.
- [6] Forbes. <http://www.forbes.com/>.
- [7] City Mayors. <http://www.citymayors.com/>.
- [8] Academic Ranking of World Universities. <http://www.arwu.org/>.

Toward Automatic Expertise Identification of Blogger

Chia Chun SHIH, Jay STU, Wen-Tai HSIEH, Wei Shen LAI,
Shih-Chun CHOU and Tse-Ming TSAI

*Institute for Information Industry, 8F., No.133, Sec. 4, Minsheng E. Rd.,
Songshan District, Taipei City 105, Taiwan (R.O.C.)
{chiachun, jaxxstu, wentai, acespot, benchou, eric}@iii.org.tw*

Abstract. The architecture of participation and sharing that encourage users to add value is one of the fundamental characteristics of a successful Web 2.0 application. Blog, as a personal publish platform on the web, removes the intermediation for channel selection thus everyone can represent himself/herself without any filtering mechanism. In this research, we present a methodology to derive user's degree of expertise from blog data and conduct an experiment using data collected from a enterprise blog system. The result shows that the average precision reaches around 0.8 and which factor is useful in our proposed method.

Keywords. Blog, Expertise location, Tagging

1. Introduction

Blog, as a personal publish platform on the web, removes the intermediation for channel selection thus everyone can represent himself/herself without any filtering mechanism. Thus based on the assumption: "a blog is a natural representation of oneself", blogs or even blogosphere could be wonderful sources to derive user preferences or even user profiles.

In this research, we attempt to derive blogger's degree of expertise in several domains. The derived expertise information could be used in many applications, such as *blog content ranking*, we can rank the experts in a blogosphere; *personalized recommendation*, content and users can be aligned to a concept space, therefore content recommendation is possible; *social navigation*, With the help of interest score and expertise score, we can easily match users with mutual interests in specific domain; and *affiliate marketing*, we can tailor the advertising strategy within the bloggers who have expertise.

This rest of the paper is organized as follows: In Section 2 we cover related works. Our methodology to induce expertise degree based on blog mining is presented in Section 3, and the experiment results shows in Section 4. Section 5 concludes our work.

2. Related Work

2.1. Blog Mining for Personalization

It has been a promising academic research direction since WWW emerged to mine personal profile from web data. Due to the lack of unified personal identity, however, the profile mining tasks were restricted to single or limited number of community/commercial sites which only catch limited aspect of a person. Recently, the prevalence of blogs [1] creates a great opportunity for web personalization. In most circumstance, content and behavior in a blog reflect explicit/implicit traits of a person. Here we review several papers about personalization based on blog mining.

Ni et al proposed a method to identify Chinese bloggers' interests from blog posts [2]. They cast the interest identification problem as a document classification problem and identify one of the most important task is to find the posts that do not represent users' interests. Another characteristic of Ni et al's approach is the support of hierarchical classification. The accuracy rate is between 16.5% to 44.5%, compared to human inspection. The objective of [3] is to identify bloggers' interests as well, and however, it considers more features than [2]. In [3], textual, temporal, and interactive features are taken into account. The results are compared to the declared interests categories in user profile to calculate the accuracy. The results show that different features have different impact in each interest categories, but average speaking textual feature is still the most useful feature. The work by Mishne and Rijke in [4] is to generate recommended book categories for bloggers through analysis of blog content. The work can be separated into two parts: locating the indicators in blog content and matching indicators with actual products. The results are compared to the categories of Amazon recommended books. The best accuracy rate is 31%.

2.2. Expertise

There are different definitions about "expertise" in previous works. We want to figure out how they define "expertise" and how they evaluate it.

[5] analyzes tagging behavior in a social bookmarking service and builds a social network around the clustered tag space. The clusters form different expertise areas. And a modified PageRank algorithm is performed to define the expert ranking. [6] wants to find optimal search algorithm like PageRank and HITS to find experts in social networks. The expertise here equals authorship in a linkage structure. [7] tries to build a personal profile that contains all the research areas. It also builds evolutionary expertise models using literature citations to match researchers' roadmap. Through the model, we can look into the relationships of researchers' master disciplines. The expertise here is the quality and quantity of publications [8] proposed a recommender system to recommend experts and show how it works and why it is beneficial. It typically performs in an enterprise and help users to address the experts in the organization. The expertise here is the working experience and employee's skills. [9] wants to find an expert in a domain specific forum such like a java forum. The expertise is defined by the answering relationships in different posts. It proposes an ExpertiseRank algorithm to evaluate the expertise level. A user who answers more questions is an expert. [10] is a community service provided by Yahoo!. It provides a platform for users to ask questions and give answers. The user who gives more answers those askers feel satisfied he will gain more rating points. The expertise here is the points that a user gain from the

askers. [11] describes a way to extract social network from e-mail activity. The expertise level here is also determined by the linkage algorithm like HITS.

According to the above previous works, we can conclude three main definitions of “Expertise”: Expertise derived from Question-Answering mechanism [9,10], expertise derived from authorship [5,6,11] and expertise derived from work experience [7,8].

2.3. Concept Space Construction from Folksonomy

A folksonomy is an Internet-based information retrieval method consisting of collaboratively generated, open-ended labels that categorize content such as Web pages, online photographs and Web links [12]. Vander Wal coined the term folksonomy: “A folksonomy is the result of personal free tagging of information and objects (anything with a URL) on the internet for one’s own retrieval. The tagging is performed in a social environment (shared and open to others). The tagging action is done by the person consuming the information” [13]. In Collaborative tagging environment, two different relations between any pair of tags can be defined by aggregating tagging data [14,15]. First, two tags are in a “Co-Resource” relation if they are adopted for the same resource. This relation is stronger between tags with more shared resources. Second, two tags used by the same user are in a “Co-User” relation. While the “Co-Resource” relation is most appropriate for establishing a public concept hierarchy, the “Co-User” Relation is most suitable for establishing a private concept hierarchy.

Concept similarity computation is divided into four steps: collecting the tagging data; deriving the similarity of tags; maintaining the similarities, and employing the concept space for some services. At the data collecting stage, most related works obtained tagging data from the largest collaborative tagging web application, del.icio.us [16,17,21]. Some others constructed their own systems [13] for testing, or accumulated data from other web applications. In the second step, most investigations determined the tag similarity [17,18,22,23]; some others performed clustering on tags [17,19,25], while others utilized facet or ontology [19,24]. The personalization topic is most common in the third step [23]. In the application stage, most studies attempted to improve the search and ranking services [17,20,21], while others.

3. Proposed Method

We attempt to derive user expertise level using semantic and interaction cues in blogs. Figure 1 explains our thought. Two kinds of user data, tag usages and user interaction records are gathered to compute the interest/expertise degree of each user. We’ll explain the calculation details in next three sections.

3.1. Tag Usage Vector

Tagging is a de-facto typical feature in latest web applications. Unlike traditional hierarchical classification scheme, which only allows assigning limited number pre-defined categories to an item, tagging allows free-style item annotation which matches the diversity characteristic of Web. In blog platforms, tags are widely used by bloggers to annotate authored articles which help organize articles.

Based on the assumption that tags are nature reflection of users’ perspective and authored articles represent users’ interests/expertise, we view the set of tags (i.e. Tag-

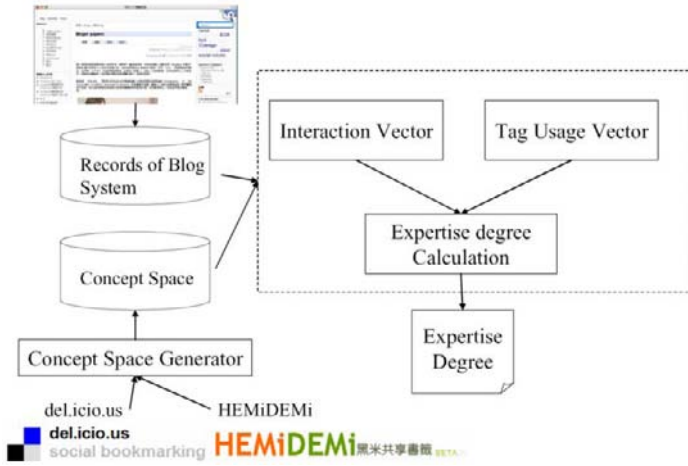


Figure 1. Method Architecture.

clouds) in blogs as a self-declared interest/expertise profile. More use of tags implies higher interest/expertise, and vice versa. In summary, we'll calculate tag frequency of all articles in each blog and generalize a tag usage vector for each blogger. The tag usage vector will be used to calculate the interest/expertise degree of each user.

Formally, we specify the tag usage vector of user u as S_u :

$$S_u = f(t_1, u) / f(t_1), f(t_2, u) / f(t_2), \dots, f(t_n, u) / f(t_n), \quad (1)$$

where $f(t, u)$ corresponds to the frequency of use of tag t_1 by u and $f(t)$ corresponds to the frequency of use of tag t_1 by all users. Each element in S_u will be a number in the range of $[0, 1]$ and represent the normalized frequency of use of each tag.

3.2. Interaction Vector

User Relationship Modeling

Blog is not only a publication platform but also an interaction media. There are various clues that relationships between bloggers can be explicitly or implicitly detected in blogosphere. The explicit clue is to be the blogroll (frequently-read blog list) in the sidebar of blogs, however, blogrolls are often updated periodically and may not reflect up-to-date relationships. The implicit clue is embed in blog articles, including comments, trackbacks, and article links. Comments and article links are very popular in blogosphere. Trackbacks are less popular than comments and links, but widely-used by experienced bloggers. Comments are the most popular way to express opinion in blogosphere. Web surfers leave instant feedbacks to blog articles through comments. If surfers have their own blogs, their blog urls are often been referred in comments as a part of contacts. Trackbacks are another means to express opinion to an article. Trackbacks could be seen as a special kind of comments, where the comments are so verbose that the commenter chooses to publish in his/her own blog. And article links are the hyperlinks to other resources presented in the article content.

Since these relationships are built upon tagged articles, we could model relationship as the follows:

If an interaction behavior (here, we refer to comments, trackbacks, or links) B is initiated by user U1 (who comments, trackbacks or authored a link in articles) to another user U2 on article A (the article being commented, trackbacked, or linked), the relationship R built in this interaction is modeled as:

$$R = (U1, U2, B, T(A)), \quad (2)$$

where $T(A)$ corresponds to the tag set of the article A and is used to identify the background of the interaction.

For ease of use in later analysis, we decompose each relationship R as atomic relation(AR):

For each relationship $R = (U1, U2, B, T(A))$, we decompose R into $\{AR\}$, where each $AR = (U1, U2, B, t, w)$, where t is an element in $T(A)$ and w(a weight ranged from 0 to 1) is the importance of the AR in this R. (we'll explain how to calculate weight in a few sections later) Now, we get a bag of atomic relationship, which will be used to construct domain-specific interaction graph for further analysis.

Interaction Graph Construction

Based on the atomic relationships derived from blogs, we construct interaction graphs for each tag and apply graph analysis algorithm to estimate the expertise level of users.

We separate atomic relationships into groups according to the t attributes (i.e. underlying tags of the relationship) and draw an interaction graph for each group. In the graph, the nodes are users and the edges are relationship between users. The weight of an edge is the w attribute of the atomic relationships. And for simplicity, all kinds of relationship (comments, trackbacks, and links) are equally weighted.

We adopt HITS algorithm [26] to estimate the expertise level of users, because the concept of "authorities" could be naturally mapped to the concept of "expertise" in that more citations implies more authoritative and authoritative articles are usually authored by experts. HITS algorithm produces two scores, hub score and authority score, for each node. Since our main purpose is to estimate the expertise level, we only take authority score into account.

After the calculation of HITS algorithm, an interaction vector is built for each user. The interaction vector of user u (I_u) is formulated as:

$$I_u = (a(u, t1), a(u, t2), \dots, a(u, tn)), \quad (3)$$

where $a(u, tn)$ corresponds to the authority score of user u in the interaction graph of tag tn. If the authority of u in t score doesn't exist, the $a(u, t)$ is equal to zero.

3.3. Expertise Degree Calculation

Eventually, we use the following formula to calculate expertise degree of user u (E_u):

$$E_u = \alpha * S_u + (1 - \alpha) I_u, \quad (4)$$

where α is an empirically tuned parameter.

3.4. Concept Space Generator (Aggregate Tags into Clusters)

Long-tail is a very ordinary phenomenon in UGC(User-Generated Content)-based applications. We found that over 80% of tags used less than three times in our dataset. If we only rely on tags to filter domain-specific data, it is very likely to produce inaccurate results due to the lack of input data (tag usage, relationships) in many domains. Therefore, we introduce “concept space” to aggregate tags into clusters. Each cluster is composed of one or many tags so that the amount of input data would be more likely to be enough for analysis.

Figure 2 presents the design of the concept space generator, which analyzes the tag space and establishes a weighted concept relation network among tags. The priory algorithm [21] is adopted to determine the concept distance. Figure 3 also shows an example of how to construct a weighted concept relation network.

First, support count for each tag is derived from the tagging table to establish a C1 table. The candidates are then filtered according to the threshold (in this case the threshold is 2) to build an L1 table. Third, by joining L1 table, tag-pairs are generated and store the data in C2 table. Fourth, we the C2 table is filtered out using the threshold employed in the second step. Finally, the confidence value of tags A – B is calculated with the following equation:

$$conf(A \Rightarrow B) = P(B | A) = \frac{Sup(A \cup B)}{Sup(A)} \quad (5)$$

Relations of Tag A – B is equal to confidence of $A \Rightarrow B$. The priory algorithm derives it from divide the support count of A and B to the support count of A. After the concept space construction, we can obtain the associated tags with any tag.

A problem occurs due to the inclusion of concept space. We call it as the “Duplicate weight” problem. Here is an example. Assume the concept space puts the three tags into a category (photography): photography, camera, and pictures. And a user publishes an article tagged with the three tags and another user b comments on this article.

Following the computation process we mentioned before, a relationship (a, b, comment, (photography, camera, pictures)) is built and then the relationship is decomposed into three atomic relationships: (a, b, comment, photography), (a, b, comment, camera), and (a, b, comment, picture). Because these atomic relationships are grouped into the same category (photography), all atomic relationships will be count in the HITS calculation of this category. The result implies: if an interaction occurs in articles tagged with more similar tags, it would be more possibly to be over-counted. It doesn't make sense.

For the problem, we provide a solution with the help of concept space. The main idea of this solution is to decrease the importance of mutually related tags in a relationship. Before going ahead to the solution details, we firstly express the idea by the above example.

Roughly speaking, the solution is composed of two main steps: 1) find the most general tags in a relationship, 2) assign a weight to each tags in the relationship. We'll explain the process using the above example.

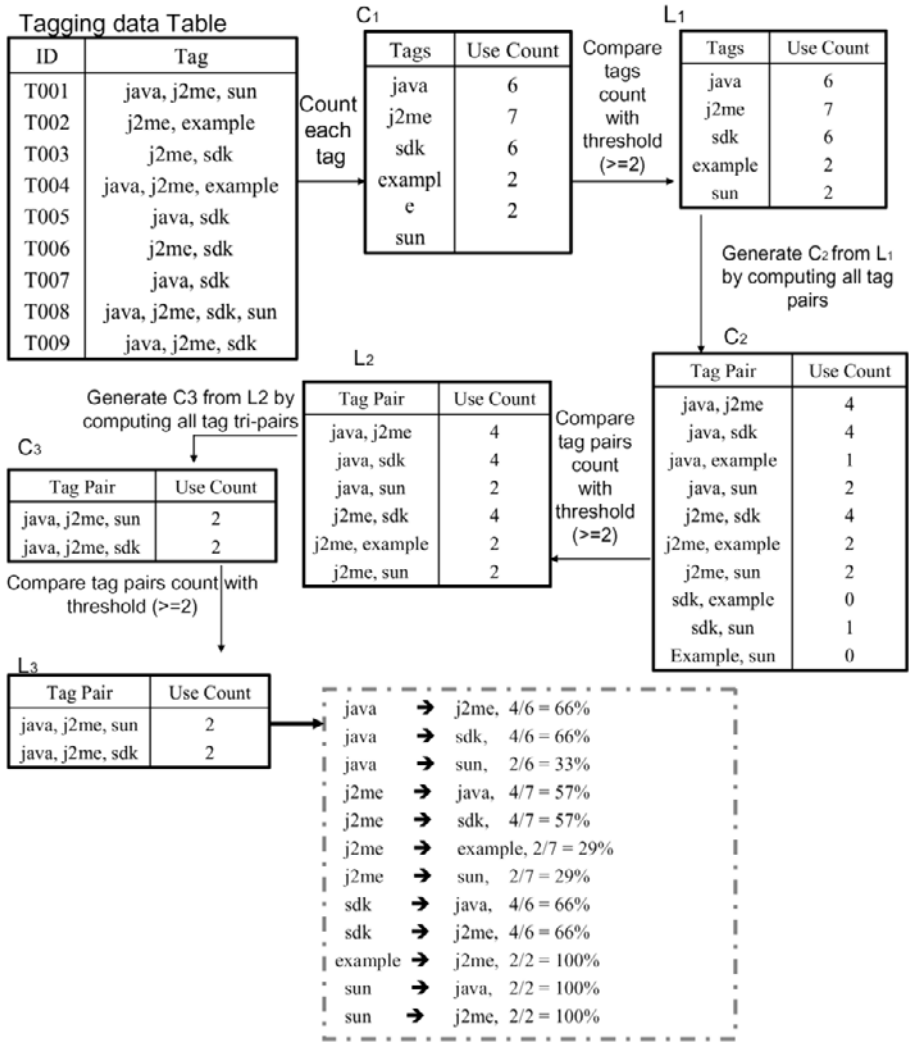


Figure 2. Concept Space Generating.

Assume the concept space shows the degree of relationship as the following:

Table 1. The degree of relationship

	Photography	Camera	Pictures
Photography	–	0.62	0.73
Camera	0.50	–	0.57
Pictures	0.52	0.42	–

Table 2. The sum of the degree of the relationship

	Photography	Camera	Pictures	Sum
Photography	–	0.62	0.73	1.35
Camera	0.50	–	0.57	1.07
Pictures	0.52	0.42	–	0.94

The value in the matrix is the degree of relationship from row item to column item, for instance, the degree of relationship from “photography” to “camera” is 0.62.

First, we would like to find the most general tag from the three. We define the most general tag in a relation as “the tag with the highest degree of relationships with the others in a relationship”. Hence, we calculate the sum of each row, and select the tag with the largest sum as the most general tag. In this example, “photography” is selected as the most general tag. (See the following table.)

Then, all terms are compared to the most general tag to determine weights. The weight of a tag t will be $(1 - (\text{the degree of relationship from the most general term to } t))$. This implies that the more related to the most general term, the less the weight. The following table shows the result of this example.

When we decompose relationships into atomic relationships, the weight will be shipped into an attribute w of atomic relationships. The weight will be a dumping factor when authority value and hub value are reciprocated in HITS algorithm.

Let’s back to the example. Originally, the relationship counts three times, but now, it counts $1.65(1 + 0.38 + 0.27)$ times, that would be more close to what we want.

4. Evaluation

4.1. Experiment Setting

We collect the data from a CMS (content management system) in an organization. We construct a blogosphere and every user can tag his articles. There are 42 users, 415 articles and 440 tags. The language in the articles is Chinese. The users’ background is the employees in the organization. We determine four domains to evaluate our method. The four domains are “Web 2.0”, “e-learning”, “sport” and “movie”. The reason to choose these four domains (tags) is the richness of this topic and the degree of human understanding. Table 4 presents the four topic and theirs tag cluster. The method presented in Section 3 is performed and our algorithm recommends 3 to 6 experts in different domains. Then we ask 14 judges to rank these recommended experts. They are all the users in the community and they judge the rank by their points of view to the recommended experts.

4.2. Evaluation Index

We use the precision concept in IR (information retrieval) to evaluate our results. The precision definition to our evaluation is:

Table 3. The result of the example

	Photography	Camera	Pictures
Photography	–	0.62	0.73
Weight	1	0.38	0.27

Table 4. Clusters of four domains

Domain cluster	Related tags
Web2.0	Mashup research 線上社群(online community) social computing social networking openpne Collaborative tagging folksonomy blog learning2.0 e-learning rss social bookmark ads2.0 ads 廣告(advertisement) 廣告 2.0(advertisement 2.0)
e-learning	web2.0 learning2.0
運動(sport)	運動傷害(sports injury)
電影(movie)	音樂(music)

Precision =

$$\frac{|\{\text{human-defined experts}\} \cap \{\text{recommended experts}\}|}{|\{\text{recommended experts}\}|}$$

(6)

The $p@1$ means that the precision of the first rank and $p@3$ means that the precision of top 3 ranks. For example, if the algorithm recommends user A, user B, user C are the top 3 experts in the domain. But the judges think that user A, user C and user D are the top 3 experts. The $p@3$ is $2/3 = 0.667$. In the following section we use average precision to evaluate the proposed method:

Average precision =

$$\frac{\sum_{i=1}^n precision@i}{n}$$

(7)

4.3. Result of Experts Recommendation Evaluation

We set α , the parameter of expertise degree of user u (E_u), to 0.5 and 0.7. After the calculation, it generates the ranks in four domains. We compare our ranks with the judges' ranks and the result is listed below.

The average precisions in four domains are around 0.8, it shows that our method is useful to determine domain experts compared to judges' opinion. We can see that the effectiveness to add clustering factor is slight. When $\alpha = 0.7$, the result is more stable. We will make some discussion in the following section.

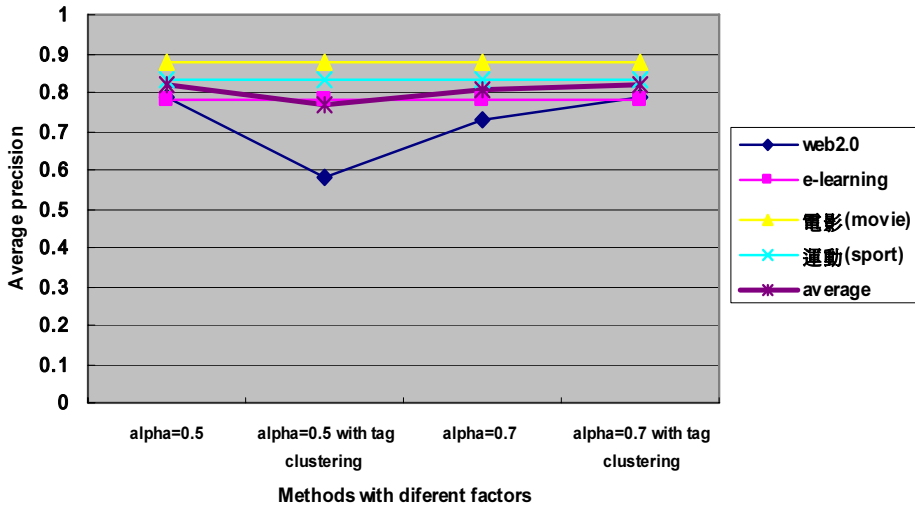


Figure 3. The Evaluation Result.

4.4. Discussion

Tag Usage is More Useful

The average precision when $\alpha = 0.7$ (0.8125) is better than $\alpha = 0.5$ (0.7934), which means tag usage vector is more useful than interaction vector. Although when α is bigger, the result is better, we can't directly say that the interaction vector is useless. The judges may first usually see his articles are relevant to the domain or not. Then they might see the numbers of his articles. The tag usage vector represents the above meanings in certain degree. The interaction vector may discover the relationship that human can't observe directly because human usually observe the blog by the content not the links. The judges' behavior might be a reason that tag usage vector is more useful than interaction vector in the experiment. The tag usage vector is important but the interaction vector is to enforce or verify the expertise level.

Tag Clustering Influence

Because the tags in the CMS system aren't rich enough, the clustering result doesn't seem very well. "web2.0" is the core cluster in the tag set. It covers lots of other tags. "Movie" has a relation with "music"; clearly the result isn't very reasonable. The purpose to do the clustering is to enhance to describe a blogger's domain more precisely. If the clustering result is more reasonable, we may obtain a better result. The algorithm won't plus some unnecessary scores from irrelevant tags via reasonable tag clusters.

5. Conclusion and Future Work

This paper wants to discover the experts in a blogosphere. We determine the expertise level by both semantic and interaction cues in blogs. We also use the tags to model the two cues. Generally speaking, tags present the domains and topics information to us and we can find the domain experts by tags. The result of our method is acceptable.



Figure 4. A personal concept graph.

The average precision is around 0.8 in our simple evaluation. It also shows that tag usage is more useful. We hope we can do more evaluation in a larger community to verify our method.

It is possible to develop different applications based on our method. Our first plan is to present the derived user expertise information in a personal concept graph (See Fig. 4.), in which nodes represent the expertise domain of the user and edges represent the relationships between domains. Historical evolution of personal concept graph is also shown to help users monitor changes of expertise. The following figure will be part of user profile in our enterprise CMS. We believe that a graph-based representation would help user easily navigate others' expertise domain.

References

- [1] The State of the Live Web, April 2007 (<http://technorati.com/weblog/2007/04/328.html>).
- [2] Xiaochuan Ni, Xiaoyuan Wu, Yong Yu, Automatic Identification of Chinese Weblogger's Interests Based on Text Classification, WI'06.
- [3] Chun-Yuan Teng and Hsin-Hsi Chen, Detection of Bloggers' Interests: Using Textual, Temporal, and Interactive Features, WI'06.
- [4] Gilad Mishne and Maarten de Rijke, Deriving Wishlists from Blogs – Show us your Blog, and We'll Tell you What Books to Buy, WWW'06.
- [5] J. Alan and D. Seligmann, Collaborative tagging and expertise in the enterprise, WWW'2006.
- [6] J. Zhang and M. Ackerman, Searching for expertise in social networks: a simulation of potential strategies, GROUP'2005.
- [7] X. Song, B. Tseng, C.-Y. Lin and M.-T. Sun, ExpertiseNet: relational and evolutionary expert modeling, 10th International Conference on User Modeling, 2005.
- [8] D. cDonald and M. Ackerman, Expertise Recommender: a flexible recommendation system and architecture, CSCW'00.
- [9] Jun Zhang Mark S. Ackerman and Lada Adamic. Expertise Networks in Online Communities: Structure and Algorithms, WWW'2007.
- [10] Yahoo! knowledge+ <http://tw.knowledge.yahoo.com/>.
- [11] Christopher S. Campbell, Paul P. Magilo, Alex Cozzi and Byron Dom. Expertise Identification using Email Communications, CIKM'2003.
- [12] WikipediaTM: <http://en.wikipedia.org/>.
- [13] Thomas Vander Wal folksonomy presentation at Online Information Conference in London: <http://www.online-information.co.uk/oi05/day2.html> (2005).

- [14] Ulises Ali Mejias, Tag Literacy http://ideant.typepad.com/ideant/2005/04/tag_literacy.html (2005).
- [15] Scott A. Golder, Bernardo A. Huberman, The Structure of Collaborative Tagging. Systems Tech Report Of Information Dynamics Lab, HP Labs (2005).
- [16] C. Cattuto, V. Loreto, L. Pietronero, "Collaborative Tagging and Semiotic Dynamics", 2006.
- [17] G. Begelman, P. Keller, F. Smadja, "Automated Tag Clustering: Improving search and exploration in the tag space", WWW2006, Edinburgh, 2006.
- [18] C. Schmitz, A. Hotho, R. J  aschk, G. Stumme, "Mining Association Rules in Folksonomies." IFCS, Ljubljana, 2006.
- [19] P. Schmitz, "Inducing ontology from Flickr tags", WWW2006, Edinburgh, 2006.
- [20] C. Schmitz, A. Hotho, R. J  aschk, G. Stumme, "Information Retrieval in Folksonomies: Search and Ranking", the 3rd European Semantic Web Conference, volume 4011 of Lecture Notes in Computer Science, Budva, 2006.
- [21] P. Schmitz, "Trend Detection in Folksonomies", the 1st Conference on Semantics And Digital Media Technology, Athens, 2006.
- [22] M. Aurnhammer, L. Steels, P. Hanappe, "Integrating collaborative tagging and emergent semantics for image retrieval", WWW2006, Edinburgh, 2006.
- [23] A. Byde, H. Wan, S. Cayzer, "Personalized Tag Recommendations via Tagging and Content-based Similarity Metrics", ICWSM, Colorado, 2007.
- [24] Alexxandre Passant, "Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs", ICWSM, Colorado, 2007.
- [25] Ajita John and Dor  e Seligmann, "Collaborative Tagging and Expertise in the enterprise." WWW2006, Edinburgh, 2006.
- [26] J. Kleinberg. Authoritative sources in a hyperlinked environment. In Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms, pages 668-677, ACM Press, New York, 1998.

Managing Co-reference Knowledge for Data Integration

Carlo MEGHINI ^a, Martin DOERR ^b and Nicolas SPYRATOS ^c

^a *CNR – ISTI, Pisa, Italy*

^b *FORTH – ICS, Crete, Greece*

^c *Université Paris-Sud – LRI, Orsay Cedex, France*

Abstract. This paper presents a novel model of co-reference knowledge, which is based on the distinction of (i) a model of a common reality, (ii) a model of an agent's opinion about reality, and (iii) a model of agents' opinions if they talk about the same object or not. Thereby it is for the first time possible to describe consistently the evolution of the agent's knowledge and its relative consistency, and to formally study algorithms for managing co-reference knowledge between multiple agents if they have the potential to lead to higher states of consistency, independent from the particular mechanism to recognize co-reference. As an example, a scalable algorithm is presented based on monitoring atomic knowledge increments and an abstract notion of belief ranking. The presented approach has a wide potential to study the formal properties of current methods to find co-reference, and to lead to new methods for the global management of co-reference knowledge.

Keywords. Co-reference, knowledge service, algorithms

Introduction

Information integration is traditionally done by reconciling data coming from different sources under a common schema. This is a well-known approach in heterogeneous database integration, which recently also involves the use of formal ontologies. Integration of data under a common schema allows for accessing information about the same kinds of things and same kinds of relations as defined by the integrated schema.

Our work relies on a different aspect of information integration. Independently if data are brought under a common schema or the sources operate in their own independent ways, and we simply try to connect them into a coherent network of knowledge. The two basic requirements for such a network to be created and managed are the following:

1. Decide whether two data elements residing in different sources refer to the same thing. [15] describes this relationship between data elements as the co-reference relation.
2. Maintain persistent co-reference relation and make it accessible to the individual sources (agents) participating in the network.

We note that co-reference between two data elements is independent of their nature or whether they belong to the same source. Actually, the problem exists even if the data

elements are not classified under any schema. For instance, historical documents fall under no particular schema whatsoever and yet establishing co-reference between text elements in two documents is fundamental for historians in order to reconstruct history.

Indeed, if we regard the instantiation relation between data elements and respective schema elements they belong to as yet another association of data elements, schema integration can also be regarded as the co-reference problem to match the referred schema elements. Therefore co-reference recognition may even be regarded as the most elementary operation of information integration.

Curiously enough, the problem of establishing and maintaining a co-reference relation has been addressed so far only in some symptomatic ways, as described in more detail in Section 1. There are some methods to assist finding co-reference: Typically, after integrating databases on the schema level, various heuristics are employed to remove “duplicate” entries coming from different sources and referring to the same thing. On the other hand, librarians invest a lot of time to create large name spaces, such as thesauri, “authority files” of person names, or gazetteers of place-names, in the hope that people will use the names in these spaces as identifiers in their local sources. Scientists and scholars all over the world maintain card boxes or create dedicated indices in books to store co-reference knowledge. Hypertext links and other “see also” links may indirectly encode co-reference knowledge. These methods suffer either from precision or scalability. OWL introduces the relationship “same as”, without exploring the theoretical question in which way two nodes can be distinct and still be “same as” at the same time.

There is no systematic literature about the epistemological nature of co-reference knowledge itself, i.e., what are the relations between the knowledge of different agents and reality, and what are the possible inconsistencies, how can they be detected and reduced. Consequently, there is no widely accepted concept of long-term preservation and management of co-reference knowledge itself. Rather than dealing with ways to identify co-reference, this paper simply assumes that there *are* ways of varying reliability to identify co-reference or non-co-reference relations and deals with the fundamental problem what to do with this knowledge in a distributed environment, in order to increase local and global consistency and the degree of knowledge sharing. To do so, we present a novel model of co-reference knowledge, which is based on the epistemological distinction of (i) a model of a common reality, (ii) a model of an agent’s opinion about reality, and (iii) a model of agents’ opinions if they talk about the same object or not. Thereby it is for the first time possible to describe consistently the evolution of the agent’s knowledge and its relative consistency, and to formally study algorithms for managing co-reference knowledge between multiple agents if they have the potential to lead to higher states of consistency, independent from the particular mechanism to recognize co-reference. As an example, a scalable algorithm is presented based on monitoring atomic knowledge increments and an abstract notion of belief ranking.

In this paper, we only interested in whether or not a co-reference relation is assumed. The idea is that the established co-reference information is *preserved* and can be used to collaboratively build large-scale networks of knowledge. Our theory is motivated by situations that occur between research or business groups maintaining huge, consolidated information records, with potentially billions of identifiers for things, people, places etc. We are not interested in the odds of natural language interpretation. Co-reference clusters, i.e. the set of links being found to connect references to the same object, are normally very small. Most things in the world are rarely referred to. However, some things,

such as Goethe or Paris, may be referred to many many times. We assume that the topology of co-reference clusters, once established, will be similar to that of reference networks of scientific papers. Some sources are preferred as reference, so that a pattern of multiply connected “stars” of different size is the most likely one. Wrong connections between large clusters may cause an immense number of wrong inferences. Therefore efficient ways to narrow down possibly wrong connections and to resolve inconsistencies in a distributed environment are a major concern.

1. Related work

There is virtually no related work about co-reference knowledge itself. Since the problem becomes more and more urgent in the Semantic Web, just recently a paper appeared that makes a limited proposal to manage co-reference [10]. It fails however to make the necessary epistemological distinctions. A fuzzy notion of context is introduced, and synonyms within a context are “bundled”. The authors assume that coreference between different contexts may point to different things, confusing the problem of agreement if something is one or two things with the problem to know if someone speaks about the same thing or not. Therefore, they do not provide a viable method to manage co-reference knowledge. In [17], the authors recognize the fact that standardization and centralization is not necessary to manage co-reference, but they still assume that there must be a unique digital surrogate for a real world item to ensure unique meaning, whereas we show that even that is not necessary. Only a strongly distributed approach has the potential to deal with co-reference of items on the Web.

A major motivation for this paper is to describe how the current approaches to support co-reference detection can be deployed more effectively. These approaches can roughly be divided into *pro-active* and *reactive* methods:

In reactive methods, typically, data are first integrated under a common schema and then possible *duplicate* data entries are detected and merged. *Duplicate detection* methods (e.g., [3]), sometimes also found under the more general term *data cleaning* methods, employ various heuristics to (a) compare the attributes of items in different data sources, and (b) estimate the probability that they mean actually the same thing (e.g., it is reasonable to assume that two references to a person with the same name, same birth date and birth place are actually identical). Unfortunately, different sources tend to register different properties for the same things, properties of items may change over time or be unknown. It requires *pre-existing shared* knowledge about the values of the properties being compared. Actually, data cleaning methods mix three different probabilities: (a) that two different items have *accidentally* the same properties, (b) that the same properties are at all registered for the same item in different sources and (c) that these properties are registered in the same way in different sources. These factors limit the reliability of these methods, but since they are scalable, they are efficient to guide manual verification to *find co-reference* based on *further evidence*. We use in this paper the fact that there are methods providing a belief or reliability ranking in a set of co-reference links. Normally, in reactive methods, detected co-reference is *not* preserved beyond the immediate scope and purpose. We claim that these methods would be far more effective if the co-reference links they produce would be globally preserved and published for open access and future, semiautomatic validation.

In *pro-active* methods, sometimes also found under *data cleaning*, identifiers of things are normalized before integration takes place, in order to increase the chance that other sources will normalize their identifiers in precisely the same way. One way to do that is by using rules, such as the cataloging rules librarians use to encode identifiers for books or authors—the most important system being AACR [2]. Rules do not help in cases when people use pseudonyms, when books do not expose standard front pages, when historical places are known under multiple names, etc. They are more helpful to avoid false matches. Even when applying rules, encoding errors may cause identifier variants. Therefore library organizations such as OCLC (Ohio, US) offer data cleaning services that validate entries sent by libraries against a huge database of entries they know of. This method has the advantages and disadvantages of *authority files*. In a way, it is a duplicate detection process between the source to be “cleaned” and the reference source.

Librarians, scholars and scientists from many disciplines have been heavily investing in so-called authority files or better knowledge organisation systems (KOS) [18], which register names and characteristics of authors [12], historical persons [11], places ([9],[8]), books and other items, as well as categories (Library of Congress Subject Headings[5], the Art & Architecture Thesaurus[20], Unified Medical Language System UMLS[21], etc.), and associate them with a preferred, unique representation in a central resource. Preferably, the KOS should list enough properties of the described items so that a user may match this description with another description under her control, and use the KOS suggested preferred identifier as a unique identifier in her source. The method is successful as long as the preferred identifiers are globally unique, correctly applied, and the KOS does not change the representation later. It is more reliable than automated data cleaning. Unfortunately there are many KOS in use, and few maintainers, notably the projects VIAF[19], LEAF[12] for person authorities, and MACS[14] for subject headings, have understood the relevance of *managing and preserving co-reference* relations between KOS. In order to find all references made via a preferred identifier from a KOS, still a global index similar to current Web search engines would be necessary. Unfortunately, there are no efforts to do so. KOS are also used to support data mining, i.e. the so-called Named Entity Recognition [4]. They help to decide, if a word in a text might be the name of a place, person, etc.

The major draw-back of KOS-based methods is that a central authority is assumed, which makes the method non-scalable, always lagging behind application. There are *far more* particulars (persons, objects, places, events) than categories (universals), making the lack of scalability very severe for particulars. Whereas the Semantic Web research is strongly focused on matching or mapping universals (e.g.[13]), here we address situations that a priori, but not exclusively, apply to particulars. We describe methods that can be applied to managing the connection of multiple KOS by co-reference links.

A variant of the *pro-active* methods is the *referent tracking* suggested by [22]. The idea is, that in controlled environments, such as health-care institutions, things like patients, body-parts, blood and tissue samples are multiply referred to in different documentation systems accompanying diagnostic and therapeutic processes. The authors suggest methods and a good practice to make sure that the notion of identity is not lost between the different documents pertaining to related processes.

Akaishi et al. [16] describe a statistical information retrieval method, in which they trace related documents via characteristic co-occurrences of terms common to docu-

ments. In a way, co-occurrence of terms can be regarded as signature of relations. Hence the method may be regarded to pertain to co-reference of relations.

Common to all current methods is that they provide partial, isolated solutions. Non does really manage co-reference knowledge once it is found. In this paper, we present a simple, but effective epistemological model to manage co-reference knowledge. We assume a community of collaborating agents whose communications relies on co-reference relations. These relations are built on the basis of bilateral agreements, and maintained in a decentralized manner, similarly to what is postulated in “emergent semantics” [1]. They are consolidated into an ever-growing, coherent system of meaning by virtue of a few global rules. We thereby implicitly show that a central authority and exhaustive description of items is not necessary, as useful it might be. All existing automated methods to detect co-reference may be used to support or enhance the respective agreement processes.

2. Language structures and co-reference relations

We assume a countable, non-empty set of objects Obj and a collection \mathcal{A} of $n > 2$ agents A_1, \dots, A_n who speak about Obj . An agent may represent a whole community of speakers that share some knowledge about Obj . For example, an agent might represent the community of users of some source that stores information about Obj . For the purposes of this paper, however, we do not need to make the distinction between users of the source and the agents representing them. We assume that each agent A_i is endowed with:

- a *vocabulary* V_i that is a non-empty, countable set of identifiers which the agent uses to refer to (or denote) objects in Obj ;
- a *reference function* R_i associating each identifier in V_i with some object in Obj .

Without loss of generality we can think of an identifier as a kind of URI (Universal Resource Identifier), independent from who knows its meaning. The only requirement is that its encoded form is different from the encoded form of any other identifier. In practice, this can be achieved by a mechanism adding a name-scope identifier to the encoded form of each local identifier. For example, `French:diffusion` is regarded as distinct from `English:diffusion`. The successful worldwide management of domain names shows that there is no reason to question the feasibility of assigning unique domain names (*i.e.*, unique agent names).

We make the following assumptions:

Assumption 1

The sets V_1, \dots, V_n are all distinct and pairwise disjoint; we will let \mathcal{V} stand for the collection vocabularies (*i.e.*, $\mathcal{V} = \{V_1, \dots, V_n\}$).

Assumption 2

Each reference function R_i is:

- *total*, that is each identifier of vocabulary V_i denotes some object, for $i = 1, \dots, n$;

$Obj = \{1, 2, 3\}$	R_1	R_2	R_3
$V_1 = \{i_1, i_2\}$	$\begin{array}{c c} i_1 & 1 \end{array}$	$\begin{array}{c c} j_1 & 1 \end{array}$	$\begin{array}{c c} k_1 & 1 \end{array}$
$V_2 = \{j_1, j_2\}$	$\begin{array}{c c} i_2 & 2 \end{array}$	$\begin{array}{c c} j_2 & 3 \end{array}$	$\begin{array}{c c} k_2 & 2 \end{array}$
$V_3 = \{k_1, k_2, k_3\}$			$\begin{array}{c c} k_3 & 3 \end{array}$

Figure 1. Elements of a language structure

- *injective*, that is no two identifiers from the same vocabulary denote the same object; in other words, vocabularies are to be understood as, *e.g. authority files* in libraries, each offering “official names” for the domain entities; it is crucial that each such name be unique within the language; of course, any official name can have a number of synonyms which the users of the language may use at their will;
- *private*, that is accessible *only* to agent A_i .

We will let \mathcal{R} stand for the collection of reference functions (*i.e.*, $\mathcal{R} = \{R_1, \dots, R_n\}$).

Assumption 3

Every object is denoted by at least one identifier of some vocabulary; in other words, there is no object unknown to all agents.

A *language structure* λ is a 4-tuple $\lambda = \langle Obj, \mathcal{A}, \mathcal{V}, \mathcal{R} \rangle$ satisfying all the above assumptions. The *vocabulary* of λ is denoted by \mathcal{L}_λ (or simply by \mathcal{L} when no ambiguity may arise) and it is defined as follows:

$$\mathcal{L} = \bigcup \mathcal{V} = V_1 \cup \dots \cup V_n$$

Figure 1 shows some elements of a language structure, consisting of three agents A_1 to A_3 which speak about a domain of three objects, represented as the first three positive integers. Throughout the paper, we will use this language structure as our running example.

In a language structure λ , it may happen that two identifiers from different vocabularies, say $x \in V_i$ and $y \in V_j$, refer to the same object, that is $R_i(x) = R_j(y)$. In this case, we say that x and y *co-refer*. In formal terms, this defines a relation \approx_λ (or simply \approx) over identifiers as follows:

$$x \approx y \text{ iff } R_i(x) = R_j(y).$$

We shall refer to this relation as the *co-reference* relation of the language structure. It is important to realize the difference between co-reference and synonymy: co-reference holds between identifiers of *different languages*, whereas synonymy holds between an identifier in V_i and any number of synonyms from the associated synonym set S_i .

It is easy to see that the co-reference relation is an equivalence relation, which induces a partition $[\mathcal{L}]$ on \mathcal{L} , given by:

$$[\mathcal{L}] = \{ [i] \mid i \in \mathcal{L} \}.$$

Each block $[i]$ of $[\mathcal{L}]$ consists of the identifiers denoting the same object as i . It follows from Assumption 3, that there exists a bi-jection between Obj and $[\mathcal{L}]$, associating every object $o \in Obj$ with the block of the identifiers denoting o . This association captures

\approx^r		
i_1	j_1	$1 \longleftrightarrow \{i_1, j_1, k_1\}$
j_1	k_1	$2 \longleftrightarrow \{i_2, k_2\}$
i_2	k_2	$3 \longleftrightarrow \{j_2, k_3\}$
j_2	k_3	

Figure 2. A (reduced) co-reference relation and its equivalence classes

naming, therefore we will call the block $[i]$ of identifiers denoting the object $o \in Obj$ as the *names* of o ; in the opposite direction, we will say that o is *named* by $[i]$. As a consequence of Assumption 2 (injectivity of reference functions), the names of an object come *each* from a different vocabulary. Formally, for all vocabularies V_i and identifiers $x_1, x_2 \in V_i$,

$$x_1 \neq x_2 \text{ implies } [x_1] \neq [x_2] \quad (1)$$

In proof, $[x_1] = [x_2]$ implies $R_i(x_1) = R_i(x_2)$ and therefore $x_1 = x_2$. Consequently, each block has at most as many identifiers as vocabularies: $|[x]| \leq |\mathcal{V}|$. If a vocabulary V_j has no identifier in the block $[x]$, then V_j has no name for the object named by $[x]$. Another way of saying this, is to say that if x and y are co-referring identifiers from different vocabularies V_i and V_j , then x does not co-refer with any other identifier in V_j :

$$x \in V_i, y, y' \in V_j, V_j \neq V_i, x \approx y, y' \neq y \text{ imply } x \not\approx y' \quad (2)$$

(1) and (2) are easily seen to be equivalent, but the latter highlights the fact that co-reference has implicit negative facts, in addition to the positive ones.

As a last remark, it is easy to verify that the complement of co-reference (with respect to $\mathcal{V} \times \mathcal{V}$), $\not\approx$ is irreflexive and symmetric.

Figure 2 shows the co-reference relation and its equivalence classes in our running example. To simplify the presentation, we only show the a reduced \approx (*i.e.*, we do not show reflexive, symmetric or transitive pairs), denoted as \approx^r . For clarity, equivalence classes are shown in association with the object they name.

3. Sharing co-reference knowledge

We view the language structure λ as being the “reality” under study. Initially, each agent A_i knows part of the reality, namely his own vocabulary V_i and reference function R_i as well as synonym set S_i and relation syn_i . The knowledge of the agent increases when he engages in communication with another agent A_j . This communication can be direct (or synchronous) if it takes places when both the agents are present, or indirect (asynchronous) if it happens through the exchange of documents. During communication, agent A_i uses an identifier, say x , in his own vocabulary V_i to refer to an object in Obj . But this creates a problem, because A_j must be able to *de-reference* x , that is to determine what is the object under discussion (*i.e.*, the object $R_i(x)$). However, R_i is only accessible to A_i , thus A_j is left with an undoable task.

In order to overcome this problem, we envisage a service, called *co-reference knowledge service*, to which A_j can *ask* which identifiers are known to co-refer with x . If, amongst the returned identifiers, A_j finds one in his own language, say y , then A_j is able

<i>co</i>		<i>nco</i>	
\dot{i}_1	\dot{j}_1	\dot{j}_2	k_2
\dot{j}_1	k_1		

Figure 3. Tables of a co-reference structure

to identify the referenced object by applying to y his reference function R_j . If no identifier in V_j is returned by the service, then the object named by x is unknown to agent A_j , and no de-reference is possible. In this case, A_j may ask the identifiers which are known not to co-refer with x , thus being able to determine which objects are *not* named by x .

The question arises how the co-reference knowledge service can acquire the knowledge it is asked about. The answer is that the service *is told* this knowledge by agents, as a result of *negotiations*. A negotiation involves two agents A_i and A_j and two identifiers in their respective vocabularies, say $x \in V_i$ and $y \in V_j$, and aims at ascertaining whether x and y co-refer. A negotiation may have one of the following outcomes:

- The agents agree that x and y co-refer.
- The agents agree that x and y do not co-refer.
- The agents are not able to reach an agreement.

In the first two cases, agents tell the service the found relationship between identifiers, thus increasing the global knowledge. In the latter case, no knowledge is gained, thus nothing is told to the service. It is important to notice that negotiations are supposed to take place *externally* to the service, which gives no support to their making. Rather, the service manages the outcomes of negotiations, whether positive or negative, in order to make communication successful according to the schema outlined above.

In this schema, the co-reference knowledge service must support the following operations:

- `tell-co(i, j)`, by means of which the user tells the service that i and j are known to co-refer;
- `tell-nco(i, j)`, by means of which the user tells the service that i and j are known not to co-refer;
- `ask-co(i)`, for asking the service the identifiers that are known to co-refer with i ;
- `ask-nco(i)`, for asking the identifiers that are known not to co-refer with i .

In addition, we include operations for retracting co-reference knowledge, which are necessary due to the possible incorrectness of agents in negotiations:

- `untell-co(i, j)`, for retracting that i is known to co-refer with j ;
- `untell-nco(i, j)`, for retracting that i is known not to co-refer with j ;

In order to be able to perform these operations, the service relies on a *co-reference knowledge structure* \mathcal{K} which we define as a triple $\mathcal{K} = \langle \lambda, co, nco \rangle$ where:

- λ is a language structure;
- *co* is the *co-reference table*, a binary relation over \mathcal{L} storing the pairs that are found to co-refer;

- *nco* is the *non co-reference table*, a binary relation over \mathcal{L} storing the pairs that are found not to co-refer.

The co-reference tables for our running example are shown in Figure 3. We assume that a co-reference structure is accessible by all agents.

In the following, we illustrate how a co-reference structure can be used to specify a semantics for the operations defined above.

3.1. Semantics of ask operations

Co-reference tables hold the *explicit* knowledge agreed upon by the agents during negotiations. However, there is also *implicit* knowledge; in our example, the fact that i_1 co-refers with j_1 and j_1 with k_1 is explicit knowledge, from which we can infer by the transitivity of co-reference (see Section 2) that i_1 and k_1 co-refer, or by the symmetry of co-reference that j_1 and i_1 co-refer; and maybe other. All this is implicit co-reference knowledge, which the user expects the service to be able to discover and serve in response to ask operations. Therefore, in defining the semantics of ask operations, we must take into account what is implicitly known from the explicit facts stored in a co-reference structure.

To this end, we first observe that a co-reference structure can be legitimately said to be such, only if it exhibits the basic properties of co-reference highlighted in Section 2. In order to capture this requirement, we introduce *co-reference models*: A co-reference structure $\mathcal{K} = \langle \lambda, co, nco \rangle$ is a *co-reference model* (or simply *model* for short) iff it satisfies the following conditions:

- (c1) *co* is an equivalence relation;
- (c2) for every pair $(i, j) \in co$ such that $i \in V_i, j \in V_j$ and $V_i \neq V_j$, there is a set of pairs $(i, j') \in nco$, where $j' \in V_j$ and $j \neq j'$;
- (c3) *nco* is symmetric;
- (c4) *co* and *nco* are disjoint.

Conditions (c1)-(c3) simply state the properties which characterize co-reference and non-co-reference, while (c4) adds the requirement that *co* and *nco* be disjoint. The next step is to apply these properties to a co-reference structure \mathcal{K} , thus completing the explicit knowledge in \mathcal{K} with the underlying implicit knowledge. The closure operation does precisely this. The *closure* of a co-reference structure $\mathcal{K} = \langle \lambda, co, nco \rangle$, is a co-reference structure $\mathcal{K}^* = \langle \lambda, co^*, nco^* \rangle$, where:

- co^* is the smallest equivalence relation containing *co*, thus satisfies condition (c1) above. co^* can be formally defined as follows. For any set of pairs X , we let:

$$\begin{aligned} \rho(X) &= \{(x, x) \mid (\exists y)(x, y) \in X \vee (y, x) \in X\} \\ \sigma(X) &= \{(x, y) \mid (y, x) \in X\} \\ \tau(X) &= \{(x, y) \mid (\exists z)(x, z) \in X \wedge (z, y) \in \tau(X)\} \\ f(X) &= \rho(X) \cup \sigma(X) \cup \tau(X) \end{aligned}$$

The first three functions realize, respectively, the reflexive, symmetric and transitive closure of the given argument (the specification of the last function can be

made more explicit, but we omit these details for brevity). Finally, f includes the results of these functions by taking their union. Now, for a co-reference structure $\mathcal{K} = \langle \lambda, co, nco \rangle$, define the *domain* of \mathcal{K} to be the set $\mathcal{D}_{\mathcal{K}} = \{co \cup A \mid A \subseteq (\mathcal{L} \times \mathcal{L})\}$. It can be easily verified that $(\mathcal{D}_{\mathcal{K}}, \subseteq)$ is a complete lattice having co as least element. Since $X \subseteq \tau(X)$, f is monotonic (hence continuous) on this lattice. Thus, by the Knaster-Tarski theorem, $co^* = f^n(co)$, for n finite. As a consequence, co^* exists, is unique and finite, since at each application of the f function only a finite number of pairs are added.

- $nco^* = X \cup \sigma(X)$ where

$$X = nco \cup \{(i, j) \in V_i \times V_j \mid (i, j') \in co^*, V_i \neq V_j, j' \in V_j \text{ and } j \neq j'\}$$

Existence, uniqueness and finiteness of nco^* immediately follow from those of co^* .

Resuming our example, the pair (j_1, i_1) ends up in co^* because it is symmetric to the co pair (i_1, j_1) . The same does the pair (k_1, i_1) , as it is symmetric to the pair (i_1, k_1) which can be obtained by transitivity from co .

On the other hand, the closure of the non-co-reference table adds to nco the pairs required for satisfying condition (c2) (yielding X), then closing the result under symmetry in order to satisfy also condition (c3). In our example, the pair k_2 is implicitly known not to co-refer with i_1 because (i_1, k_1) are known to co-refer, thus i_1 cannot co-refer with any other identifier in V_k ; assuming it is known that k_2 is an identifier in V_k , this means that i_1 and k_2 are known not to co-refer; by the symmetry of nco , we then obtain that k_2 and i_1 are known not co-refer.

The closure of a co-reference structure \mathcal{K} embodies the explicit co-reference knowledge in \mathcal{K} and, being an equivalence relation, exhibits the behavior of co-reference. Moreover, being the smallest structure satisfying these two properties, it is a most natural candidate for query answering on \mathcal{K} . We therefore define the semantics of ask operations by viewing them as functions associating a co-reference structure with sets of identifiers, as follows:

$$\text{ask-co}(i)(\mathcal{K}) = \{j \in \mathcal{L} \mid (i, j) \in co^*\}$$

$$\text{ask-nco}(i)(\mathcal{K}) = \{j \in \mathcal{L} \mid (i, j) \in nco^*\}$$

We simplify notation by writing $\text{ask-co}(i, \mathcal{K})$ instead of $\text{ask-co}(i)(\mathcal{K})$, and the same for ask-nco .

3.1.1. Inconsistent co-reference structures

It is important to notice that in closing a co-reference structure \mathcal{K} , (c4) may be violated, so that the resulting \mathcal{K}^* is not a model. In this case, some pair (i, j) is known both to co-refer and not to co-refer. This is clearly a contradiction, thus we define \mathcal{K} a *consistent* co-reference structure iff its closure \mathcal{K}^* is a model. Accordingly, we define the cons-ch operation as a function returning the pairs causing inconsistency:

$$\text{cons-ch}(\mathcal{K}) = co^* \cap nco^*$$

If `cons-ch` returns the empty set, then the current co-reference structure is consistent. Otherwise, the user knows which pairs are causing trouble and can intervene as described later.

In our example, if we add the pair (i_1, j_1) to the *nco* table in Figure 3, we have an explicitly inconsistent co-reference structure, since the same pair shows up in the *co* table. If we add the pair (k_2, i_1) to the *co* table, we have an implicitly inconsistent co-reference structure, because the same pair shows up in the closed non-co-reference table *nco**, as shown in a previous example.

Inconsistent co-reference structures arise from negotiations whose outcome does not correctly reflect the language structure λ and, ultimately, from *unsound* agents, that is agents that may make mistakes in negotiations. The assumption that only negotiation mistakes can cause inconsistency of the co-reference structure holds, as long as the agents share an unambiguous principle of identity about the described objects, and they maintain their reference functions injective. As it turns out, this is not an unrealistic case. It appears that inconsistency of the co-reference structure is the only diagnostic we have at information system level of errors in the negotiations. If someone finds out with whatever means that two identifiers do not co-refer, this information enters the *nco* table, which may cause an inconsistency. Any consistent arrangement of false negotiations remains undetected.

3.2. Semantics of `tell` operations

The semantics of `tell` operations establish how co-reference structures evolve. Redundancy of explicit knowledge is not helpful as long as believe values are not taken into account. Therefore, an obvious requirement is minimality, in the sense that a co-reference structure should stay as simple as possible, while embodying the knowledge that it has been told. However, ignoring knowledge explicitly told by the user is not a good idea, because it may cause loss of knowledge. As an illustration, let us consider the co-reference structure shown in Figure 1. Assume that the user wants to add the knowledge that i_1 and k_1 co-refer via the operation `tell-co` (i_1, k_1) . Now this knowledge is implicit in the co-reference structure, as it can be obtained by transitivity from (i_1, j_1) and (j_1, k_1) . We could therefore be tempted to do nothing in response to the above `tell-co`. But if we did so, a successive `untell-co` (i_1, j_1) would cause the loss of the knowledge that i_1 and k_1 co-refer. We therefore give `tell` operations the following, straightforward semantics, where $\mathcal{K} = \langle \lambda, co, nco \rangle$ is the current co-reference structure:

$$\begin{aligned} \text{tell-co}(i, j, \mathcal{K}) &= \langle \lambda, co \cup \{(i, j)\}, nco \rangle \\ \text{tell-nco}(i, j, \mathcal{K}) &= \langle \lambda, co, nco \cup \{(i, j)\} \rangle \end{aligned}$$

Note that the addition of knowledge may cause an inconsistency in the co-reference structure. However, there is no guarantee that the piece of knowledge being added reflects an incorrect co-reference relationship: it might be the case that the inconsistency is caused by knowledge which has been told previously. Thus a `tell` always results in a larger co-reference structure. Clearly, good practice suggests a consistency check after each knowledge insertion.

Analogously:

$$\text{untell-co}(i, j, \mathcal{K}) = \langle \lambda, co \setminus \{(i, j)\}, nco \rangle$$

$$\text{untell-nco}(i, j, \mathcal{K}) = \langle \lambda, co, nco \setminus \{(i, j)\} \rangle$$

Notice that if the pair (i, j) is not present in co (nco , respectively), then the former (latter) operation has no effect.

4. Implementation

We now discuss the implementation of the operations introduced so far. We begin by introducing the basic data structure for the implementation of co-reference, the co-reference graph.

Given a co-reference structure $\mathcal{K} = \langle \lambda, co, nco \rangle$, the *co-reference graph* of \mathcal{K} , $G_{\mathcal{K}}$ (simply G when no ambiguity may arise), is the undirected graph $G = (\mathcal{L}, (co \cup nco))$. Arcs arising from pairs of identifiers in co will be called *co-arcs*, while arcs arising from pairs in nco will be called *nco-arcs*. Likewise, a *co-path* is a path in G including only co-reference arcs. Finally, an identifier i is *co-reachable* from an identifier j if there exists a co-path from i to j . Figure 4 shows the co-graph for the co-reference structure of the running example. For readability, nco-arcs are shown as dashed lines.

A basic property of undirected graphs that will be very useful for the sequel, is that all the nodes from the same component of such a graph are reachable from each other [6]. Since co-reachability will play a crucial role, we focus in particular on the components of the sub-graph (\mathcal{L}, co) . We denote these components as (N_i, E_i) , for $1 \leq i \leq m$. Each component (N_i, E_i) is a connected, undirected graph. It follows that two identifiers are co-reachable from each other iff they belong to the same set N_j , for some $1 \leq j \leq m$. Moreover, by construction the sets N_i 's are a partition of the set of identifiers showing up in \mathcal{K} , and when complete co-reference knowledge is reached, they coincide with the equivalence classes discussed in Section 2, each consisting of the names that an object has in the considered vocabularies. For this reason, each graph (N_i, E_i) will be called *name graph* while each N_i will be called a *name set*.

In our running example (see Figure 4), the graph has 5 components, whose name sets are given by: $\{i_1, j_1, k_1\}$, $\{i_2\}$, $\{j_2\}$, $\{k_2\}$ and $\{k_3\}$.

4.1. Implementing ask operations

The following Proposition states the basic result for ask operations.

Proposition 1 *For any co-reference structure $\mathcal{K} = \langle \lambda, co, nco \rangle$ and identifiers $i, j \in \mathcal{L}$:*

1. $j \in \text{ask-co}(i, \mathcal{K})$ iff j is co-reachable from i in G ;
2. $j \in \text{ask-nco}(i, \mathcal{K})$ iff there exist two identifiers a and b such that $a \in \text{ask-co}(i, \mathcal{K})$, $b \in \text{ask-co}(j, \mathcal{K})$ and either a and b belong to the same language, or there is a nco-arc from a to b in G .

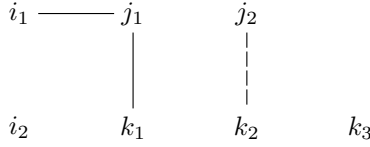


Figure 4. A co-reference graph

Proof: (1) (\rightarrow) If j is co-reachable from i in G , then $(i, j) \in co^*$ by transitivity.

(\leftarrow) We have that $co^* = f^n(co)$ for a finite n (see Section 3.1). We prove that for all $1 \leq k \leq n$, $(i, j) \in f^k(co)$ implies that j is co-reachable from i . The proof is by induction on n . For $k = 1$, $f^1(co) = co$. Thus if $(i, j) \in co$ then there is an arc from i to j in G , hence j is co-reachable from i . Suppose the thesis holds for $k = m < n$, let us prove it for $k = m + 1$. $f^{m+1}(co)$ adds to $f^m(co)$ the pairs that are obtained from those in $f^m(co)$ either by symmetry or by transitivity. In the case of symmetry, if $(i, j) \in f^{m+1}(co)$, it means that $(j, i) \in f^m(co)$, which means by the induction hypothesis that i is co-reachable from j ; then, for the symmetry of co-reachability, also j is co-reachable from i . In the case of transitivity, $(i, j) \in f^{m+1}(co)$ implies that $(i, h), (h, j) \in f^m(co)$, for some identifier h . By the induction hypothesis, h is co-reachable from i and j is co-reachable from h , which implies that j is co-reachable from i .

The proof of (2) is much simpler and is omitted for reasons of space. \square

Based on this Proposition, assuming $i \in N_k$, we have:

$$\text{ask-co}(i, \mathcal{K}) = N_k$$

$$\text{ask-nco}(i, \mathcal{K}) = \{j \in \mathcal{L} \mid (i, j) \in nco\} \cup \{j \in V_j \mid j \neq j' \in V_j \text{ and } j' \in N_k\}$$

and we may therefore conclude that both ask operations can be implemented efficiently. ask-nco requires to enumerate, for every identifier co-reachable from the given one, all the different identifiers from the same vocabulary. This enumeration may be very long, but it is the only one that correctly reflects the non-co-reference knowledge.

4.2. Detecting inconsistency

An inconsistency is caused by the same pair to be both in co^* and in nco^* . In order to devise an algorithm for eliminating the inconsistency, it is crucial to understand under which conditions a co-reference graph represents an inconsistent co-reference structure. To this end, we only need to derive the consequences of Proposition 1.

Corollary 1 For every co-reference structure $\mathcal{K} = \langle \lambda, co, nco \rangle$ and pairs of different identifiers $i, j \in \mathcal{L}$, $(i, j) \in co^* \cap nco^*$ iff i, j belong to the same name set N_k , for some k , and:

1. either N_k contains two identifiers joined by an nco-arc, or
2. N_k contains two identifiers from the same language.

It is immediate to verify that the Corollary merely combines conditions (1) and (2) of Proposition 1.

The pairs of identifiers satisfying condition 1 of the last Corollary are therefore given by:

$$C_1 = \{(i, j) \mid i \neq j, \{i, j\} \subseteq N_k \text{ and } N_k^2 \cap nco \neq \emptyset, \text{ for some } 1 \leq k \leq m\}$$

that is all pairs of identifiers which are in the same name set as two pairs that are known not to co-refer. Computing C_1 simply requires, for each pair $(i, j) \in nco$, to check whether the set N_k where one of i or j belongs, also contains the other one. If yes, then all pairs in N_k are in C_1 .

On the other hand, the pairs of identifiers satisfying condition 2 of the last Corollary are given by:

$$C_2 = \{(i, j) \mid \{i, j\} \subseteq N_k \text{ and } |N_k \cap V_h| \geq 2, \text{ for some } 1 \leq k \leq m, 1 \leq h \leq n\}$$

that is all pairs of identifiers which are in the same name set as two pairs from the same vocabulary. This requires a scanning of each name set, to find 2 identifiers from the same vocabulary. whenever such a name set is found, all pairs in it are in C_2 . Assuming each identifier carries also the identifier of the vocabulary it comes from, C_2 can be computed efficiently.

Since the pairs (i, j) satisfying one of the conditions of the Corollary are the result of the `cons-ch` operation, we have that for all co-reference structures \mathcal{K} ,

$$\text{cons-ch}(\mathcal{K}) = C_1 \cup C_2$$

and we can conclude that this operation can be efficiently implemented too.

4.3. Repairing an inconsistency

The inconsistencies found in the set C_1 arise from the fact that a pair (i, j) is both in *co* and in *nco*. In this case, all pairs of different identifiers in the same name set as i and j are in C_1 , but obviously the prime cause of the inconsistency is the pair (i, j) ; hence, the repairing will focus on this kind of pairs. One of two actions can then be performed:

- removing the (i, j) from *nco* via an `untell-nco`(i, j) operation; in this case, negotiations have brought about that the positive co-reference knowledge is to be trusted; or
- making either i or j disappear from the name set N_k where they both belong. This requires breaking all paths connecting i and j in the name graph (N_k, E_k) , by making a number of `untell-co` operations which collectively would *cut* i from j , in fact causing the name graph to split into two graphs.

Analogously, the prime causes of the inconsistencies found in the set C_2 , are those identifiers i and j that belong to the same language and also to the same name set. In order to solve these inconsistencies, therefore, one of two actions can be taken:

- to make i and j synonyms of one another, thus using one of the two as a representative of both, and recording this information somewhere¹; in practice, this would mean to substitute the removed identifier with the other one in all co-reference relationships in which it occurs; or

¹Synonym lists are commonly associated to authority files.

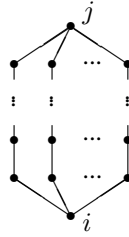


Figure 5. A co-reference graph

- to cut i from j in the co-reference graph.

Clearly, in both cases the second operation is the most difficult one, and the question arises how the co-reference service can support the user in identifying a set of wrong links making up a cut, *i.e.* the co-reference relationships that must be un-tell-ed in order to restore consistency.

The service does not have any clue as to what each identifier refers to, and the only optimality criterion that it can follow is to propose to the user *minimal* cuts, thus minimizing the number of negotiations which the user has to carry out. However, this is not a feasible strategy, because there can be an exponential number of minimal cuts. Consider for instance the graph shown in Figure 5: any set including a link from each of the paths connecting i and j is a minimal cut, and their number of such cuts is exponential in the number of paths.

In fact, even identifying *any* minimal cut is a difficult problem, which can be shown to be NP-complete.

For the hardness, Figure 6 shows a reduction from MINIMAL HITTING SET [7] to the problem of saturating the network on the right with a minimal number of links, which is the same as finding a minimal cut between i and j . A MINIMAL HITTING SET instance consists of a finite set S and a collection $\mathcal{C} = \{C_1, \dots, C_n\}$ of subsets of S . The question is whether there exists a subset X of S of minimal size, containing at least one element of any member C_i of \mathcal{C} . The network corresponding to this problem has 3 types of links:

- the links outgoing from i are one-to-one with the element of S , and so are the different nodes they lead to; the capacity of each such links is the outrank of the target node, to make sure all links outgoing from each target node are used;
- the links incoming into j are one-to-one with the members of the collection \mathcal{C} , and so are the different nodes they leave from;
- the links in between these, connect each node corresponding to an element x of S with the nodes corresponding to the elements $C_i \in \mathcal{C}$ in which x belong; each link of this and the previous type has capacity 1 to make sure it is used exactly once.

In order to saturate the network, a minimal set of links outgoing from i must be chosen, so that all links incoming into j are saturated. The way the network is built clearly guarantees that such a minimal set of links is one-to-one with a MINIMAL HITTING SET for the initial problem.

Membership in NP can be easily seen by considering that the size m of a minimal cut between two nodes of a graph can be determined in polynomial time in the size of

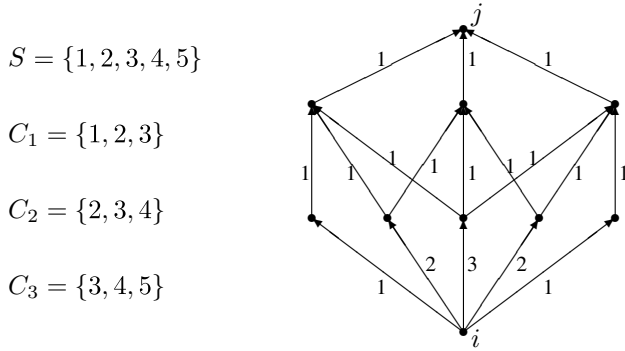


Figure 6. Reduction from MINIMAL HITTING SET

the graph. Given then a candidate solution s to the problem in question, it suffices to determine m and check whether the size of s is the same as m . If yes, then s is a minimal set of links; if not, it is not. All this can be done efficiently, thus determining a minimal set of links is an NP-complete problem.

For these reasons, we envisage an interactive and iterative strategy for cutting two identifiers in the co-reference graph. The strategy is based on the obvious observation that a cut must include a link from any path from i to j . We then envisage a $\text{repair}(i, j)$ operation, where $(i, j) \in \text{cons-ch}$, which, when issued by the user, causes the following interaction to take place:

1. The service tries to compute any co-path from i to j .
2. If no such co-path exists, then the repair operation is completed successfully. Clearly at the first iteration a co-path is always found as a consequence of the fact that $(i, j) \in \text{cons-ch}$.
3. The service shows to the user the links on the obtained co-path which the user has not yet considered (*i.e.*, re-negotiated). At the first iteration no link has been re-negotiated by the user, so all links on the co-path are shown; from the second iteration on, the user may have already negotiated some links, which need not therefore be shown to her again.
4. The user, via negotiations, identifies at least one, possibly many of the shown links as incorrect and untell-co each one of them.
5. The services executes each untell-co operation issued by the user and iterates.

In order to efficiently execute its part of the above protocol, the co-reference service must be able to quickly determine whether i and j are connected in the co-graph (step 1), and, if so, serve the links on a co-path to the user, discarding the ones already considered in previous iterations (step 3). One way of doing so, is to maintain a *spanning tree* of each name graph (N_k, E_k) . A spanning tree of a graph is a minimal set of edges of the graph that connect all nodes. By definition, in a tree there is exactly one path from any two nodes, so the spanning tree is cycle-free and therefore very suitable for the task at hand.

The next Section outlines algorithms and data structures for implementing the co-reference service, based on the results presented so far.

4.4. Algorithms and data structures

The basic data structure for co-reference is a table C with four columns, each holding the information about a co-reference pair; in particular,

- the first two columns are used to store, respectively, the first and second element of the co-reference pair;
- the third column stores an identifier of the name set where the co-reference pair belongs;
- the last column stores a binary value v indicating whether ($v = 1$) or not ($v = 0$) the co-arc representing the pair belongs to the spanning tree of the name graph.

The data in the last 2 columns do not represent any entity in the formal model, they are just used for implementation purposes. Non-co-reference information is stored in a two column relation N , each row of which keeps a non-coreference pair.

In what follows, we will sketch the execution of each operation in the interface of the co-reference service, by analyzing the possible scenarios and outlining the actions to be taken. The following invariants regarding the table C can be easily proved, for instance by induction on the size of C :

- *Co-arc uniqueness.* The first two columns are a key of C , in any order (i.e., C has two keys). Moreover, the projection of C on the first two columns is an asymmetric relation. Overall, this means that $(i, j, x, y) \in C$ implies that for no $z \neq x$ and $w \neq y$, $(i, j, z, w) \in C$ or $(j, i, z, w) \in C$. For simplicity, we call a “ ij tuple” any tuple that has i and j (in any order) in the first two columns.
- *Weak C_2 consistency.* $(i, j, x, y) \in C$ implies that i and j do not belong to the same language. This does not mean that C is C_2 -consistent, but simply that “evident” inconsistencies are avoided by maintaining a synonym list for each identifier. We do not enter into the details of how these lists are implemented.
- *Name consistency.* Language and name set identifiers are consistent: it cannot happen that the same language identifier is associated with two different name set identifiers in two different tuples of the C table. More technically, there is a functional dependency from either of the first two columns of C to the third. Based on this dependency, we use the notation $\nu(i)$ to indicate the identifier of the name set where i belongs.

For brevity, we only consider `tell` operations, and conclude with a remark on how to efficiently perform the first step of a `repair`, aiming at finding a path from two given identifiers from the same name set.

Figure 7 presents the `tell-co` procedure. `tell-co` takes as input two identifiers i and j and if there already exists an ij tuple in C , it does nothing. Otherwise,

- If both i and j do not occur in C (line 2), then: if i and j belong to the same language, then in order to maintain the weak C_2 consistency, i is chosen to be the official representative of both, having the other as a synonym; if i and j belong to two different languages, then the insertion of the tuple $(i, j, A, 1)$ into C means that a new name set is created, having A as identifier and including both i and j ; in addition, (i, j) is made part of the spanning tree of A .

procedure tell-co(i, j : identifiers)

```

1.  if no  $ij$  tuple exists in  $C$  then
2.    if neither  $i$  nor  $j$  occur in any tuple in  $C$  then
3.      if  $i$  and  $j$  belong to the same language then  $\text{syn}(i) \leftarrow \{j\}$ 
4.      else insert( $i, j, A, 1$ ) into  $C$  where  $A$  is a new name set identifier
5.    else if  $j$  does not occur in any tuple in  $C$  then
6.      if  $i$  and  $j$  belong to the same language then  $\text{syn}(i) \leftarrow \text{syn}(i) \cup \{j\}$ 
7.      else insert( $i, j, \nu(i), 1$ ) into  $C$ 
8.    else
9.      if  $i$  and  $j$  belong to the same language then begin
10.        replace  $\nu(i)$  by  $A$  in  $C$  where  $A$  is a new name set identifier
11.        replace  $\nu(j)$  by  $A$  in  $C$ 
12.        replace  $j$  by  $i$  in  $C$ 
13.         $\text{syn}(i) \leftarrow \text{syn}(i) \cup \{j\}$ 
14.      end
15.    else
16.      if  $i$  and  $j$  belong to the same name set  $A$ , then insert( $i, j, A, 0$ ) into  $C$ 
17.      else begin
18.        replace  $\nu(i)$  by  $A$  in  $C$  where  $A$  is a new name set identifier
19.        replace  $\nu(j)$  by  $A$  in  $C$ 
20.        insert( $i, j, A, 1$ ) into  $C$ 
21.      end

```

Figure 7. The tell-co procedure

- If one of i and j (say j) does not occur in C , there can be two cases: (1) i and j belong to the same language (line 6), in which case j is stored as a synonym of i . (2) i and j belong to two different languages (line 7), then the fact that j co-refers with i is recorded by inserting $(i, j, \nu(i), 1)$ into C , along with the fact that (i, j) is part of the spanning tree of $\nu(i)$.
- If both i and j occur in C , there can be the same two cases: (1) i and j belong to the same language (line 9); in this case, it follows from the weak C_2 consistency of C that i and j belong to two different name sets. Then, in term of the co-graph, i and j are coaleshed into a single node representing one of the two identifiers, having the other one as a synonym. This is implemented by (a) replacing both $\nu(i)$ and $\nu(j)$ by a new name set identifier A in C ; (b) replacing j by i in C ; and finally (c) storing j as a synonym of i . (2) i and j belong to two different languages; in this case, if i and j belong to the same name set A , then $(i, j, A, 0)$ is inserted into C since i and j are already connected by the spanning tree of A and therefore the newly added co-arc does not have to be on the spanning tree. If i and j belong to two different name sets, then tell-co connects two whole name sets, as follows: (1) a new name set identifier A replaces both $\nu(i)$ and $\nu(j)$ in C ; (2) $(i, j, A, 1)$ is inserted into C .

Figure 8 presents the untell-co procedure. untell-co takes as input two identifiers i and j which belong to different languages by the C_2 -consistency of C . If no ij tuple exists in C , nothing is done. Else, let (i, j, A, v) be such a tuple, unique by co-arc uniqueness.

procedure untell-co(i, j : identifiers)

```

1.  if there exists an  $ij$  tuple  $(i, j, A, v)$  in  $C$  then
2.    if  $v = 0$  then remove  $(i, j, A, v)$  from  $C$ 
3.    else begin
4.       $N_i \leftarrow \{k \mid (i, k) \text{ is a path in the spanning tree of } A \text{ without the } (i, j) \text{ co-arc}\}$ 
5.       $N_j \leftarrow \{k \mid (j, k) \text{ is a path in the spanning tree of } A \text{ without the } (i, j) \text{ co-arc}\}$ 
6.      if no tuple  $(x_i, x_j, X, b)$  exists in  $C$  where  $x_i \in N_i$  and  $x_j \in N_j$  then begin
7.        remove  $(i, j, A, v)$  from  $C$ 
8.        replace each tuple  $(x, y, X, c)$  in  $C$  such that  $x, y \in N_i$ , by  $(x, y, A_i, c)$ 
           where  $A_i$  is a new name set identifier for  $N_i$ 
9.        replace each tuple  $(x, y, X, c)$  in  $C$  such that  $x, y \in N_j$ , by  $(x, y, A_j, jc)$ 
           where  $A_j$  is a new name set identifier for  $N_j$ 
10.     end
11.    else
12.      for each tuple  $(x, y, A, 0)$  in  $C$  such that  $x \in N_i$  and  $y \in N_j$  do begin
13.        remove  $(i, j, A, 1)$  from  $C$ 
14.        replace the tuple  $(x, y, A, 0)$  in  $C$  by  $(x, y, A, 1)$ 
15.      end
16.    end

```

Figure 8. The untell-co procedure

- If $v = 0$, then the corresponding co-arc (i, j) is not on the spanning tree of the name set A , hence the tuple is removed from C .
- If $v = 1$, N_i (respectively, N_j) is the set of identifiers reachable from i (j) in the spanning tree of A without the (i, j) co-arc. By definition of spanning tree, N_i and N_j are a partition of the name set where i and j belong. There can be two cases: (1) if no tuple exists in C having x_i and x_j in the first two columns, where $x_i \in N_i$ and $x_j \in N_j$, then (i, j) is the only co-arc connecting i and j ; in this case two different name sets are created with the remaining co-arcs in the name set A (lines 7-9). (2) If $(x, y, A, 0)$ is any tuple in C such that $x \in N_i$ and $y \in N_j$, it is replaced by $(x, y, A, 1)$ and the tuple $(i, j, A, 1)$ is deleted from C .

The procedures for operating on non-coreference are quite straightforward. In particular:

- tell-nco(i, j) does nothing if an ij tuple exists already in N . Otherwise, the tuple (i, j) is added to N .
- untell-nco(i, j) does nothing if no ij tuple exists in N . Otherwise, the tuple (i, j) is removed from N .

Finally, the crucial step in repair(i, j) is the identification of the path on the spanning tree leading from i to j . By using a simple backtracking strategy, this requires connecting the co-arcs corresponding to the tuples in C marked with a 1 in the fourth column, starting from those who have an i as one of the two ends, until one is found which as j as one of the two ends.

Conclusions

To the end of establishing a necessary service for data integration, we have analyzed the notion of co-reference as it stems from language structures, that is vocabularies of identifiers with reference functions assigning an object to every identifiers. We have then defined the primitive operations of a service for maintaining co-reference knowledge. These operations reflect a functional approach to knowledge representation [15], including tell operations, by which users can tell the service the result of negotiations, and ask operations, by which users can extract implicit and explicit co-reference knowledge from the service, based on what they previously told the service. The semantics of these operations has been defined, and an efficient implementation has been derived, based on the co-reference graph.

References

- [1] Karl Aberer, Philippe Cudré-Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand-Said Hacid, Arantza Illarramendi, Vipul Kashyap, Mecella Massimo, Eduardo Mena, Erich J. Neuhold, Olga De Troyer, Thomas Risse, Monica Scannapieco, Félix Saltor, Luca De Santis, Stefano Spaccapietra, Steffen Staab, and Rudi Studer. Emergent semantics principles and issues. In *DASFAA*, page 25 38, 2004.
- [2] Chartered Institute of Library American Library Association, Canadian Library Association and Information Professionals (Great Britain). *Anglo-American Cataloguing Rules 2004: Binder Inserts (Loose Leaf)*. American Library Association, 2004.
- [3] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, 2003.
- [4] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics, 2006.
- [5] Lois Mai Chan. *Library of Congress Subject Headings: Principles and Application Fourth Edition*. Library and Information Science Text Series. Libraries Unlimited, April 2005 2005.
- [6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. The MIT Press, 2nd edition, 2001.
- [7] Michael R. Garey and David S. Johnson. “*Computers and Intractability: A Guide to the Theory of NP-Completeness*”. W. H. Freeman and Company, New York, 1979.
- [8] Patricia Harpring. Proper words in proper places: The thesaurus of geographic names. *MDA Information*, 2(3), 5-12., 2(3):5 12, 1997.
- [9] L. L. Hill and Q. Zheng. Indirect geospatial referencing through placenames in the digital library: Alexandria digital library experience with developing and implementing gazetteers. In *ASIS1999*, 1999.
- [10] A. Jaffri, H. Glaser, and I. Millard. Uri identity management for semantic web data integration and linkage. Submitted to 3rd International Workshop On Scalable Semantic Web Knowledge Base Systems, Vilamoura, Algarve, Portugal, 2007.
- [11] M.Baca et al. J.M.Bower. *Union List of Artist Names - A User's Guide to the Authority Reference Tool, Version 1.0*, Getty Art Information Program. G.K.Hall, New York, 1994.
- [12] M. Kaiser, H.-J. Lieder, K. Majcen, and Vallant H. New ways of sharing and using authority information. *D-Lib Magazine*, 9(11), 2003.
- [13] Y. Kalfoglou and W. M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1 31, 2003.
- [14] Patrice Landry. The macs project : Multilingual access to subjects (lcsh, rameau, swd). *International cataloguing and bibliographic control*, 30(3):46 49, 2001.

- [15] H.J Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155 212, 1984.
- [16] Ken Satoh Mina Akaishi, Koichi Hori. Topic tracer: a visualization tool for quick reference of stories embedded in document set. In *IV2006*, page 101 106, 2006.
- [17] Bijan Parsia and Peter F. Patel-Schneider. Meaning and the semantic web. In *WWW2004*, page 306, 2004.
- [18] Manjula Patel, Traugott Koch, Martin Doerr, and Chrisa Tsinaraki. D5.3.1: Semantic interoperability in digital library systems. Deliverable, Project no.507618, DELOS, A Network of Excellence on Digital Libraries, June 2005.
- [19] Edward T. O'neill Rick Bennett, Christina Hengel-Dittrich and Barbara B. Tillett. Vial (virtual international authority file): Linking die deutsche bibliothek and library of congress name authority files. In *WLIC2006*, 2006.
- [20] Dagobert Soergel. The art and architecture thesaurus (aat): a critical appraisal. *Visual Resources*, 10(4):369 400, 1995.
- [21] National Institutes of Health United States National Library of Medicine. Unified medical language system, 2007. accessed March 6, 2007.
- [22] Barry Smith Werner Ceusters. Strategies for referent tracking in electronic health records. *Journal of Biomedical Informatics*, 39(3):362 378, 2006.

On Reducing Relationships to Property Ascriptions

Jørgen FISCHER NILSSON

DTU Informatics, The Technical University of Denmark, Denmark

Abstract. This paper discusses whether relationships can preferably be reduced to properties in conceptual modelling and formal ontology engineering. The discussion takes as point of departure the subject-predicate form of logic supporting a monadistic view.

Keywords. Entity/relationship model, relationships and properties, monadistic view, formal ontologies, description logic

1. Introduction

Models of reality for information processing systems generally apply as central notions those of classes of particulars (entities, individuals), and properties, and relationships of classes and particulars. Thus the well-known entity-relationship models (ER-models) [1,2] comprise

- classes (of instances), and
- attributed properties to instances of the classes
- (mainly binary) relations between classes.

The question addressed in this paper is whether relationships can preferably be reduced to properties in conceptual modelling and formal ontology engineering. If the stated question can be answered affirmatively, it means that the third main component mentioned above can be exchanged in favor of adorned properties.

1.1. Relationships

Given classes A and B a relation R between entities or instances (relata) belonging to these (not necessarily distinct) classes is often depicted as

$$A \xrightarrow{R} B$$

This triple, consisting of a relation-labelled arc between two classes, forms the basic component in a conceptual modelling diagram. The arrow is to indicate the order of arguments in the relation rather than suggesting a functional property. In general the considered relations are many:many (m:n).

Given any pair of classes of particulars, there, of course, always in general exists many relations in the mathematical sense of sets of pairs of entities. However, a con-

ceptual modelling enterprise addresses identification of relevant relationships (possibly or actually) existing *in reality* rather than solely *in a mathematical realm*. This becomes apparent if it is claimed in a particular context, say, that there are no relations between two given (non-empty) classes.

Metaphysically, relationships are more abstract than the usually tangible particulars with accompanying properties (tropes) forming classes in a conceptual model. Therefore, now, the question is under which circumstances a relationship as the above may preferably be reshaped as a property ascription to (instances of) the involved classes? Suggestively in diagram form we have

$$A \longrightarrow (R : B)$$

informally understood as ascription of a property to instances of A . A property is formed by the would-become attribute R (quality, determinable) and a value slot from B (quale, determinate). If this recasting is possible as well as desirable, further the question arises as to whether it is then to be accompanied by the reciprocal property assignment

$$B \longrightarrow (R^{-1} : A)$$

At first sight the relations might simply be re-conceived as functions (maps) taking maps as arguments through the well-known currying process. However, the ontological aspect calls for a more comprehensive analysis.

1.2. Formal Ontologies

The considered dispensing with relationships in favor of property ascription is actualized by the recent widespread interest in formal ontologies. Ontologies are basically classification models structured by the class inclusion relation *isa* giving rise to the notion of (immediate) superclass C'' of an (immediate) subclass C' as in the customary diagram component

$$\begin{array}{c} C'' \\ \uparrow \\ C' \end{array}$$

The inclusion relation enjoys a special status since it is a relation between classes as such rather than a relation between instances of classes, contrast R above. This is evident from the presence of the inclusion relation at various places throughout in conceptual model diagrams. Similarly, for mereological part-of relations. Apart from the class inclusion relation and possibly also the mereological relations, it seems that other relationships especially in ontologies might be replaced by property ascription as suggested above. Such ascribed properties are usually thought of as being inherited downwards to all subclasses of the ontology except in case of the so-called non-monotonic inheritance.

1.3. Aspects

The possible re-conception or reshaping of relationships as properties has some interrelated aspects:

1. The issue can be understood as whether ascribed properties possess a more firm existential metaphysical status than relations. This metaphysical aspect is touched further in the following sect. 2.
2. The issue can be understood as a pragmatic problem in the conceptualization and specification and engineering of information processing systems. As such it may involve circumstances such as the functional properties of the involved relations as known from data base modelling.
3. The issue can also be understood as addressing the mathematical or computational properties of the logical systems favored in contemporary formal modelling and ontology building.

It is easy to confuse these aspects not the least since they seem to be related.

2. Some Historical Background Remarks

In the context of formal modelling the issue of relations versus properties is bound up with the applied representation language and logic. Here one should recall the dichotomy between algebraic logics in the tradition of Boole and Peirce versus the logical calculus with quantified variables due to Frege. In [3] Russell at p. 221 writes

It is a common opinion – [...] – that all propositions ultimately consist of subject and predicate.

Then he explains ways of dealing with relational propositions in accordance with the view just quoted:

Given, say, the proposition aRb , where R is some relation, the monadistic view will analyse this into two propositions, which we may call ar_1 and br_2 , which give to a and b respectively adjectives supposed to be together equivalent to R .

This monadistic view Russell attributes to Leibniz, cf. [4], see also further [5].

Following this view the simple example proposition "pancreas produces insulin" is to be replaced by two subject-predicate propositions, say: "pancreas is insulin-producing" and "insulin is pancreas-produced", where we have introduced two somewhat artificial extra-lexical predicates serving as adjectives for a subject.

At first sight such a rewriting may be viewed as a step backwards from the successful development and use of predicate calculus throughout the 20th century. It might be accused of being a regression to a pre-Fregean logic insisting that propositions are shaped as "Subject is Predicate" as in Aristotelian and Leibnizean logics. However, we wish to argue that the monadistic view deserves attention especially in contemporary methodologies for ontology development.

3. A Logical Analysis of Relations vs. Properties

Let us discuss and analyze relationships and properties in the framework of predicate calculus. Consider again

$$A \xrightarrow{R} B$$

with relation R and related classes A and B .

In predicate logic n -argument relations standardly become n -ary predicates, with classes becoming unary predicates. Suppose that an entity/particular (named) a is R -related to particular b . This gives rise to the logical atomic fact

$$R(a, b)$$

If we were to use lambda-abstraction properties might be formed as $\lambda x.R(x, b)$ for a and $\lambda y.R(a, y)$ for b . Turning these relationships into property ascriptions within predicate logic can be done by way of implicational sentences. Clearly for the considered relationship, to each x there exist a y such that $R(x, y)$ and *vice versa*. Then we have

$$\forall x A(x) \rightarrow \exists y (R(x, y) \wedge B(y))$$

and conversely

$$\forall x B(x) \rightarrow \exists y (R(y, x) \wedge A(y))$$

The latter sentence can be reformulated as

$$\forall x B(x) \rightarrow \exists y (R^{-1}(x, y) \wedge A(y))$$

appealing to the inverse relation of R , which is bound to exist mathematically though it need not be lexicalized separately.

3.1. The Case of Particulars as Singleton Classes

Now particulars may be re-conceived as singleton classes (or conceived as possessing a singleton class counterpart). Thus for each particular k we may posit a predicate K with

$$\forall x (K(x) \leftrightarrow x = k)$$

In the present case we have thus at disposal

$$\forall x (A(x) \leftrightarrow x = a) \text{ and } \forall x (B(x) \leftrightarrow x = b)$$

Now it is easy to show that with these stipulations, from

$$\forall xA(x) \rightarrow \exists y(R(x,y) \wedge B(y))$$

follows $R(a,b)$ and *vice versa*.

Similarly from

$$\forall xB(x) \rightarrow \exists y(R(y,x) \wedge A(y))$$

follows $R(a,b)$ and *vice versa*. At first sight this may seem to be a contrived manner of expressing a simple relationship between two particulars. As we shall see in the next section one can devise appropriate formulations facilitating a desired monadistic breakdown of relationships into property ascriptions.

Appealing again to λ -calculus (thereby going beyond first order predicate calculus formally, if not necessarily in substance) ascribed properties might be formed as the abstraction

$$\lambda x.(\exists y(R(x,y) \wedge B(y)))$$

However, in relation to the above historical remarks one may recall that neither the device of lambda-abstraction chiefly due to Alonzo Church nor the currying principle of Curry were properly developed at the time of [3].

4. Variable-free Algebraic Logic (Term Logic) Form

The above predicate logical formulations may be couched in a algebraic variable-free form reminiscent of formulations in description logic [6]. Cf. also our [7].

The principal form suggested here is the monadistic subject-predicate form

$$A \text{ isa } \varphi$$

where A is either a particular or a class, and where φ is a class optionally ascribed a property as in

$$A \text{ isa } B \text{ [with] } R C$$

for the predicate logical

$$\forall x(A(x) \rightarrow B(x) \wedge \exists y(R(x,y) \wedge C(y)))$$

In case that the class B is absent (i.e., really, the universal class encompassing all other classes) one may prefer the more fluent wording $A \text{ hasprop } R C$ instead of $A \text{ isa } B \text{ [with] } R C$. As a special case of the latter there is

$$C' \text{ isa } C''$$

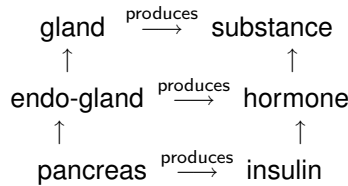
simply stating a class inclusion relationship. Recall that for the purpose of *inter alia* getting a streamlined predicate logical counterpart particulars may be re-conceived as singleton classes by way of $\forall x(K(x) \leftrightarrow x = k)$ and thus $K(k)$.

Now, with the predicate logical explication from sect. 3 a property ϕ inherits downwards in an ontology. Thus from $A \text{ isa } B$ and $B \text{ isa } \phi$ one obtains $A \text{ isa } \phi$. This is in contrast to using relational facts $R(a, b)$. Appropriate inference rules may be provided for the term logic instead of the appeal to predicate logic.

One may observe that it is not the case in general that $A \text{ isa } R C$ iff $C \text{ isa } R^{-1} A$. In [8] we analyze this logical specification language for ontologies in the setting of distributive algebraic lattice algebras.

4.1. An Example

Consider the class-relationship diagram fragment¹ for glandular organs in humans, that is organs which secrete substances such as enzymes and hormones.



One recognizes the building blocks described in sect. 1. As it appears the diagram comprises 6 classes and 1 kind of relationship in addition to the vertical *isa*-relationship. As usual upwards-pointing arcs represent the inclusion relationships between classes. Observe that even though the "produces" relation is inherited in the sense that for instance it is implied that the pancreas produces some hormone, the diagram states explicitly that pancreas more specifically produces the insulin hormone.

If pancreas and insulin are conceived as particulars, then in predicate logic we have straightforwardly `produces(pancreas,insulin)`. However, then we cannot have `produces(endo-gland,hormone)`, since the two arguments here are to be represented as unary predicates. Therefore we turn to the language from sect. 4.

In this variable-free logical language we get
 endo-gland *isa* gland produces hormone
 or strictly "atomistic"

 endo-gland *hasprop* produces hormone
 together with

 endo-gland *isa* gland.

Provided that it is to be encoded that [all] hormones are produced in [some] endo-glands we get the reciprocal

 hormone *hasprop* produced-by endo-gland

assuming here that means are put at disposal for introducing *produced-by* as the inverse relation of *produces*.

No special treatment is offered particulars such as pancreas and insulin provided that they be lifted to singleton classes as suggested. In such case we simply get

 pancreas *isa* endo-gland produces insulin

together with

 insulin *isa* hormone produced-by pancreas.

¹The diagram is understood to be fragmentary; for instance it does not tell that pancreas besides being endocrine is also an exocrine gland producing enzymes etc.

5. Reciprocals

Above we noticed that $A \text{ isa } R C$ and $C \text{ isa } R^{-1} A$ are logically independent as illustrated in the above example. We call such a pair of sentences reciprocals. Thus reciprocals unlike inverse relationships are logically independent.

In particular [13] introduces a pair of general partonomic reciprocals **has-part** and **part-for** between entity classes, appealing to an underlying part relation between individuals similar to our R above.

As an example, given that solar systems come with a planet belonging to that system, unlike moons, one has e.g. **planet part for solar-system** and **solar-system has part planet**. By contrast one has **moon part for solar-system** but not **solar-system has part moon**, again illustrating the independency of the reciprocals.

6. Conclusion

Drawing a line from the monadistic tradition in metaphysics we have argued that relationships may preferably be represented formally as property ascriptions. This is particular relevant in the context of formal ontologies shaped into classification structures enriched with property ascriptions. To this end above is devised a logical representation language in the subject-predicate form effectively subsumed by the T-box language of description logic.

As a key point here we also explain how logical relationships between particulars as found in the A-box of description logic can be reshaped in the subject-predicate form fitting ontological classifications and providing inheritance of ascribed properties.

This brings up the epistemological question of distinguishing what holds necessarily in a terminology from mere empirical facts being subjected to empirical verification and falsification, cf. also the synthetic vs. analytic distinction. This issue is also bound up with distinguishing classes and properties as such addressed e.g. in our [9].

We suggest that for use in ontologies the A-box / T-box distinction of description logic are replaced with the formulations suggested in above sections possibly prefixed with modes stating the epistemological status of the individual assertions. An insistence on ontologies comprising only necessary as opposed to contingent (inclusion) relationships, cf. Ontoclean analysis [10], leads to the issue of internal and external relations. For a discussion of the underlying essential-accidental distinction and its implications on properties see [11], the entry on relation, and [12], the entries on Bradley and on relations, internal and external.

Let us conclude these methodological remarks with a quotation from [4], which outlines an atomistic modelling analysis prevailing in ontology development as opposed to a more holistic view.

For Russell one could identify an individual of some kind A and learn all one could about it as a thing in its own right, then go on to do the same thing with an individual B , and then as a third enterprise study the relation between A and B . This third study would add to one's stock of truths, so that one could form lots of truths of the form ' A is R to B ', but this would not require abandonment or modification of what one learnt about A and B in one's original two inquiries.

References

- [1] Chen, P. P.-S.: The entity-relationship model: toward a unified view of data, *ACM Trans. on Database Systems*, 1:1, pp.9-36, 1976.
- [2] Chen, P. P.-S.: English, Chinese and ER diagrams, *Data & Knowledge Engineering*, 23, 1997. p. 5-16.
- [3] Russell, B.: *The Principles of Mathematics*, George Allen & Unwin, 1903, 1950.
- [4] Sprigge, T.: Russell and Bradley on Relations, in G.W. Roberts (ed.): *Bertrand Russell Memorial Volume*, George Allen & Unwin, 1979.
- [5] Ishiguro, H.: *Leibniz's Philosophy of Logic and Language*, Cambridge U.P., 1972, 1990.
- [6] Baader, F.: *Description Logic Handbook*, Cambridge U.P., 2002.
- [7] Bruun, H. & Nilsson, J. Fischer: Entity-Relationship Models as Grammars and Lattices - a Foundational view, *Information Modelling and Knowledge Bases XVI*, Y. Kiyoki et al. (eds.), IOS press 2005. pp. 292-301.
- [8] Bruun, H., Gehrke, M. & Nilsson, J. Fischer: Lattice-structured Ontologies, an ontological account, unpublished draft, December 2007.
- [9] Smith, B. and Rosse, C.: The Role of Foundational Relations in the Alignment of Biomedical Ontologies. In Proceedings, MedInfo 2004, San Francisco, CA. pp. pp. 444-448.
- [10] Nilsson, J. Fischer: Ontological Constitutions for Classes and Properties, 14th Int. Conf. on Conceptual Structures, H. Schaerfe, P. Hitzler, P. Øhrstrøm (eds.), *Lecture Notes in Artificial Intelligence LNAI 4068*, 2006.
- [11] Guarino, N. and Welty, C.: Evaluating ontological decisions with ONTOCLEAN. *Communications of the ACM*, 45(2):61-65, February 2002.
- [12] Audi, R.: *The Cambridge Dictionary of Philosophy*, Cambridge U.P., 1995.
- [13] Honderich, T.: *The Oxford Companion to Philosophy*, Oxford U.P., 1995.

A Domain-Specific Knowledge Space Creation Process for Semantic Associative Search

Minoru KAWAMOTO^a Yasushi KIYOKI^b

^a *Keio Research Institute at SFC, Keio University*

^b *Faculty of Environmental Information, Keio University*

Abstract. This paper presents a multiple knowledge spaces creation process for domain oriented semantic associative search. We have presented a generation method of semantic associative search spaces for domain-specific research areas. This process enables database managers and experts of the specific domain to collaborate in constructing semantic search spaces with domain-oriented knowledge. Domain-specific knowledge space creation is essentially important to obtain appropriate knowledge from expert-level document repositories. This paper presents the implementation process for knowledge space creation and the software environment for supporting collaboration work between database managers and domain experts.

1. Introduction

Various methods to retrieve domain-oriented information resources have been proposed. These methods are coming to be used widely. We have proposed a semantic associative search method[9,10] which uses a vector space model architecture representing domain specific knowledge by vector notations. We have designed a domain specific knowledge space creation process for semantic associative search[7].

The main objective of this research is to establish knowledge base creation methodology using domain-specific knowledge resources, such as textbooks and encyclopedias. This process makes it possible to obtain various semantic associative search spaces which consist of domain specific knowledge expressions. The variety of search spaces is derived from parameter settings which database manager gives in the space creation process. The parameters are set up by two values: (1) Expertise-level parameter for determining the size of a search space, (2) sub-domain parameter for determining the criteria of a search space. By setting these parameter values, it is possible to control the sizes and the objectives of the space flexibly. In our process, those two aspects label the primordial features. ‘Sub-domain parameter’ can be expressed by a single symbol. ‘Expertise-level parameter’ can be expressed by four levels corresponding to the relevance between the feature and the target sub-domain.

We have implemented two software tools which enable to cooperate with domain experts on constructing domain-specific search knowledge and to make it available in practical environment. We have performed two experiments. We clarify the effectiveness of search spaces created by our process by Experiment-1. By Experiment-2, we have discussed the workload of the space creation through our process.

By applying our process, the knowledge resources on many scientific fields can be used in the semantic knowledge search environment.

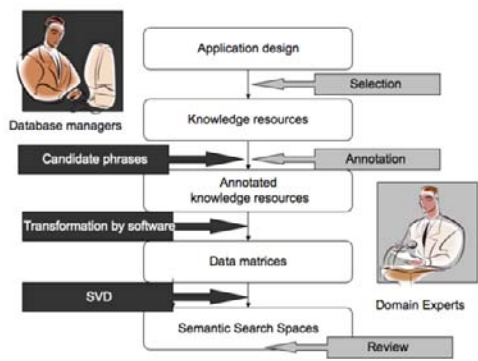


Figure 1. Collaboration architecture

2. Knowledge Space Creation

Our knowledge space creation is to build a domain-specific knowledge presentation which realizes multimedia data retrieval using experts level knowledge. By using this process, database managers and experts in a specific domain can collaborate with constructing such a knowledge presentation. The visual image of the construction process is described in Figure 2.

We have proposed a creation method of Semantic Associative Search Spaces in the past article([7]), and we clarified the feasibility of the method. The characteristic feature of the creation method exists in effectiveness of the space creation using ‘expertise level’ and ‘sub-domain’ features.

This paper focuses on the collaborative creation process by the database managers and experts in the application domain (medical doctors in the example) rather than creation methodologies themselves.

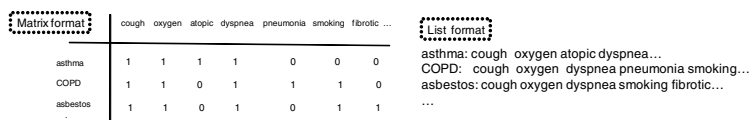


Figure 2. Data structure of the data matrices

2.1. Knowledge Expression

In our system architecture, domain-specific knowledge is formalized as featured-vector matrices. The more precise formalization of the Semantic Associative Search Method is detailed in [9,10].

In this section, we describe briefly the expression of the domain knowledge that realizes domain specific search spaces. The data structure of the domain knowledge is on vector space model. The vector spaces are obtained from data in matrix like structures which illustrate the relationships between feature phrases and key phrases. The structure of the data matrices is shown in Figure 2. The rows of the matrices denote ‘key phrases’ and the columns of the matrices denote ‘feature phrases’. As shown in the figure, the goal of the collaboration by the database managers and the domain experts is to build the matrices which can be converted into the semantic associative search spaces.

3. Constructing Process

In this section, we explain the process of the semantic associative search spaces implementation. The focus of this paper is in the Step-6 and -7.

Step-1. Application design First, the application of the search space is designed. Rest of the steps follows up the application domain.

Step-2. Selection of knowledge resources The expert of the target domain selects knowledge resources to build search spaces with. For example, textbooks of the target domain can be the knowledge resources.

Step-3. Extract the structure of the knowledge resources Extract the content structure of the knowledge resources computational assistance can be applied.

Step-4. Key-phrases preparation To build data matrices of the field, the 'key phrases' of the domain are extracted from the body text of the knowledge resources. The key phrases are typically chosen from the titles of paragraphs of the text.

Step-5. Select body texts This step extracts body texts of the textbook that reflects application design. The extraction step is made by considering the structure of the texts.

Step-6. Setup for feature-phrase candidates To reduce the workload of selecting feature phrases from the text, this process prepares candidates set of the feature-phrases.

Step-7. Markup feature-phrase with expertise level In this step, the experts markup the phrases that occurs in the body text with expertise level.

Step-8. Creation of search spaces To create search spaces with parameters which specifies target sub-domains and their sizes.

It is difficult to determine which phrases should become feature phrase, and which phrases not, except when domain experts do not assist. To ease the workload of domain experts, preparing candidate set of feature phrases and other efforts to accomplish the ease of workload are essentially important.

4. Software Implementation

We have implemented two software tools which enables construction of Semantic Associative Search Spaces from experts' knowledge and domain-specific knowledge resources such as textbooks and glossaries.

4.1. WordPicker

WordPicker is Java-based software which allows users to 'pick' and 'unpick' phrases into *feature phrases set* and determine their 'importance'. Before users start annotating phrases' importance, the candidates of feature phrases are marked up with the least importance, the default value. As to the last implementation, users classified feature phrases into importance A (the most important) to D (the least important). The screen shot of WordPicker is shown in Figure 4.

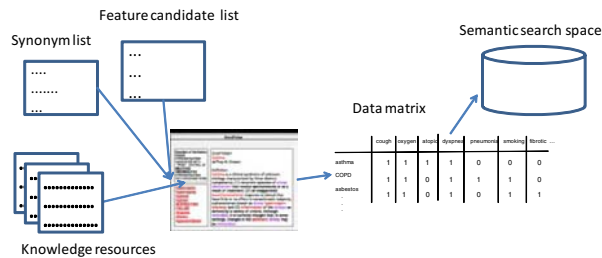


Figure 3. Implementation process using WordPicker

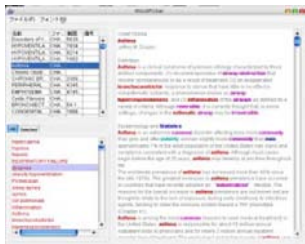


Figure 4. A screen shot of running WordPicker: Annotating

4.2. Spacemaker

Spacemaker performs creation of Semantic Associative Search Spaces[9,10] by constructing feature phrases from the feature phrases set. This software realizes automatic generation of Semantic Associative Search Spaces from source text resources and feature phrases of which the *importance* and *related sub-domains* are annotated. The annotations has been accomplished with WordPicker and former preprocessing by this phase.

Spacemaker is a command-line conversion tools built with Perl language. It outputs the feature phrase set which corresponds with the sub-domains.

4.3. Other Implementations: A Setup of feature phrases candidates

We have also achieved a non-software effort. We prepared the feature phrase candidates set for preprocessing of domain experts’ review process.

We set up feature phrase candidates for an agile feature review; word-for-word review requires too much workload. Feature phrase candidates were made from “Stedman’s Cardiovascular & Pulmonary Words: Electronic Word Book Includes Respiratory[13]”(A word list) and words appeared in the index of “Cecil Textbook of Medicine[6],”

This process is done by paying attention to prepare accurate candidates set. The accuracy of the prepared candidate set could be evaluated by the domain experts in consideration of the granularity of the phrases and the comprehensiveness. The important thing when prepare the candidates is to make exhaustive set as near a false positive one, because selecting phrases from unmarked texts requires much more workload than unselecting phrases of pre-marked phrases.

Table 1. Mappings between Sub-domains and Headlines

<i>Sub-domain</i>	<i>Title of the headline(Case insensitive)</i>		
Clinical Manifestations (CM)	clinical presentation	signs and symptoms	...
Definitions and Diagnosis (DD)	definition	definitions	...
Prevention, Treatment and Prognosis (PTP)	thrombolytic therapy	prevention, treatment and prognosis	...
Pathology and Pathophysiology (PP)	pathology of asthma	pathology and pathophysiology	...
Epidemiology (EP)	epidemiology	epidemiology and statistics	...
Etiology (ET)	etiology	incidence and prevalence	...

Table 2. Expertise level and sub-domains of feature words

<i>Feature phrase</i>	<i>Sub-domain</i>	<i>Expertise level</i>
tuberculosis	PTP, DD, ET, CM	A
chest pain	PTP, DD, PP, ET, CM	B
echocardiogram	PTP, DD	C
physiotherapy	PTP	D

5. Implementation in the medical field

We have implemented semantic associative spaces by applying our collaboration process, using software tools which are described above. We targeted the *pulmonary and respiratory* domain among the whole medical field.

We selected sub-domains together with domain-experts so that the applications using the sub-domains are respectively valuable. The sub-domains we selected are listed in Table 1. This selection was made by mapping between subsections/paragraphs title and sub-domains(n:1).

The examples of expertise levels of the feature phrases are show in Table 2. The sub-domains to which the feature phrases belong are determined by occurrence of the feature phrases in respective subsection/paragraphs.

Table 3. Sizes of matrices and Number of Dimensions by Parameters

<i>Search space ID</i>	<i>Data matrix</i>	<i>Dimensions</i>	<i>Search space ID</i>	<i>Data matrix</i>	<i>Dimensions</i>
$S_{A+B+C+D}$	131×3954	3136	$S_{A+B+C+D,DD}$	131×1668	1204
S_{A+B+C}	131×2163	1658	$S_{A+B+C,DD}$	131×855	627
S_{A+B}	131×1166	853	$S_{A+B,DD}$	131×512	328
S_A	131×903	530	$S_{A,DD}$	131×358	223

6. Experimental Study

We have performed several experiments. Experiment-1 calculates the total scores of respective search spaces. By this experiment we clarify the effectiveness of our space creation process. In Experiment-2, we examine the workload of space creation processes which consist of our approach and existing one.

Table 4. Result of Experiment-1 (*italic:* highest)

Query	Search space ID	<i>v</i>	Query	Search space ID	<i>v</i>	Query	Search space ID	<i>v</i>
Rheumatoid-factor	<i>S_{A+B,DD}</i>	643	dyspnea	<i>S_{A+B,DD}</i>	840	Cough	<i>S_{A+B,DD}</i>	753
Rheumatoid-factor	<i>S_{A+B+C,DD}</i>	619	dyspnea	<i>S_{A+B+C,DD}</i>	810	Cough	<i>S_{A+B+C,DD}</i>	722
Rheumatoid-factor	<i>S_{A+B}</i>	554	dyspnea	<i>S_{A+B}</i>	812	Cough	<i>S_{A+B}</i>	726
Rheumatoid-factor	<i>S_{A+B+C}</i>	601	dyspnea	<i>S_{A+B+C}</i>	793	Cough	<i>S_{A+B+C}</i>	756
chest-pain	<i>S_{A+B,DD}</i>	551	Smoking	<i>S_{A+B,DD}</i>	586	FEV1	<i>S_{A+B,DD}</i>	521
chest-pain	<i>S_{A+B+C,DD}</i>	576	Smoking	<i>S_{A+B+C,DD}</i>	650	FEV1	<i>S_{A+B+C,DD}</i>	499
chest-pain	<i>S_{A+B}</i>	538	Smoking	<i>S_{A+B}</i>	570	FEV1	<i>S_{A+B}</i>	434
chest-pain	<i>S_{A+B+C}</i>	573	Smoking	<i>S_{A+B+C}</i>	492	FEV1	<i>S_{A+B+C}</i>	497
Pleural-effusions	<i>S_{A+B,DD}</i>	800	Asbestos	<i>S_{A+B,DD}</i>	629			
Pleural-effusions	<i>S_{A+B+C,DD}</i>	730	Asbestos	<i>S_{A+B+C,DD}</i>	549			
Pleural-effusions	<i>S_{A+B}</i>	664	Asbestos	<i>S_{A+B}</i>	505			
Pleural-effusions	<i>S_{A+B+C}</i>	670	Asbestos	<i>S_{A+B+C}</i>	494			

6.1. Experiment-1: Quantitative Evaluation

The soundness of the obtained search spaces was examined by the quantitative experiments. We issued some sample queries and measured the accuracy of the obtained results from the search space. We have implemented the 4 semantic associative search spaces. The specifications of the spaces are shown in Table 3. $S_{A+B+C,DD}$ in Table 3, Table 4 and Figure 5 stands for the space which ‘expertise level: A and B’ and ‘sub-domain: DD’ are applied. Metadata of target data items were determined by tfidf[12]; top-ranked 10 key phrases were to be metadata of the target.

6.1.1. Experiment Evaluation

We issued some queries reflecting the application domain, ‘Definitions and Diagnosis.’ Several cases are chosen from a casebook, the keywords related to the cases were to be the queries we used. Medical doctors have evaluated the relevance score of the item.

We calculated sums of rank scores of the results performed by several queries onto each space. The ID of the search space is denoted as α , and s_r is the relevance score of the item ranked in r . The sums of scores are calculated as shown below. In this experiment, the scope where we calculated the sums of rank scores is ranged in between 1st and 20th($k = 20$).

$$v_\alpha = \sum_{i=1}^k s_i(k-i+1) \quad (1)$$

6.1.2. Result and Discussions

The result obtained from the experiment-1 are shown in Table 4 and Figure 5. As shown in the table, the spaces using our process ($S_{DD,A+B+C}$ and $S_{DD,A+B}$) made higher rank sums.

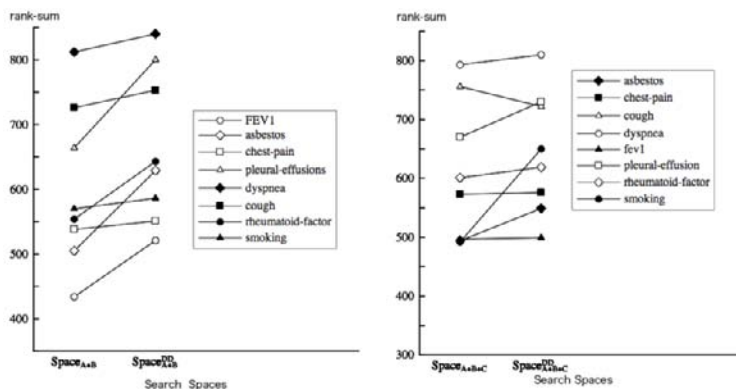


Figure 5. Result of Experiment-1

6.2. Experiment-2: Qualitative Evaluation (Thought Experiment)

We have also performed a qualitative evaluation of the collaboration processes.

We have evaluated the effectiveness of our process by comparing the difference of workloads to obtain data matrix of Semantic Search Space. We have compared the workloads between by our process and by existing approach. The point of difference is how to evaluate the term in the light of the location in the search spaces. While existing approach deals with 'Is the feature phrase candidate is to be feature phrase or not' (task t_1), our approach deals with 'Which expertise level the feature phrase candidate is' (task t_2) and 'Which sub-domain the feature phrase candidate is included in' (task t_3).

While t_1 is a task concerning about the nature of search spaces, t_2 and t_3 are tasks for which domain experts can evaluate values by their own knowledge. Therefore, the method using t_2 and t_3 is more efficient than using t_1 .

7. Related Work

Several methods and implementations have proposed to realize dimension degeneration by computation without expert-managed domain knowledge [1,3,4,5]. Our process has enabled to supervise the dimensions degeneration by domain experts. Many data retrieval using vector space model implementation have proposed [2,11]. Multimedia data retrieval methodology based on the vector space model, utilizing Semantic Associative Search Method [9,10], has applied to domain-specific multimedia data retrieval [8].

8. Conclusions

We have presented a multiple knowledge space creation process for domain-oriented semantic associative search to realize a collaborative creation environment between database managers and domain experts. This paper has also presented the implementation process for knowledge space creation and the software environment for supporting collaboration work between database managers and domain experts. We have performed

two experiments. We have clarified the effectiveness of our search space creation process, showing our experimental processes and results for actual knowledge space creation. By applying our process, we realize an effective semantic associative search environment for obtaining various knowledge resources on many scientific fields. By this work, we have enabled to build domain knowledge structure to retrieve domain-specific documents and multimedia data.

References

- [1] Rie Kubota Ando. Latent semantic space: iterative scaling improves precision of inter-document similarity measurement. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 216–223, New York, NY, USA, 2000. ACM Press.
- [2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.
- [3] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, New York, NY, USA, 2001. ACM Press.
- [4] Jr. Charles Lee Isbell and Paul Viola. Restructuring sparse high dimensional data for effective retrieval. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 480–486, Cambridge, MA, USA, 1999. MIT Press.
- [5] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [6] Lee Goldman and Dennis Ausiello, editors. *Cecil Textbook of Medicine*. W.B. Saunders Company, 22nd edition edition, December 2003.
- [7] Minoru Kawamoto, Yasushi Kiyoki, Seitaro Fujishima, and Sadakazu Aiso. A generation method of semantic associative search spaces for domain-specific documents. *IPSI Transactions on Databases*, 47(SIG 19(TOD 32)):113–126, December 2006. (In Japanese).
- [8] Minoru Kawamoto, Yasushi Kiyoki, Naofumi Yoshida, Seitaro Fujishima, and Sadakazu Aiso. An implementation of a semantic associative search space for medical document databases. In *IEEE International Symposium on Applications and the Internet (SAINT 2004) - the International Workshop on Cyberspace Technologies and Societies (IWCTS 2004)*, pages 488–493. IEEE, January 2004.
- [9] Takashi Kitagawa and Yasushi Kiyoki. A mathematical model of meaning and its application to multidatabase systems. In *Proceedings of 3rd IEEE International Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems*, pages 130–135. IEEE, April 1993.
- [10] Yasushi Kiyoki, Takashi Kitagawa, and Takanari Hayama. A metadatabase system for semantic image search by a mathematical model of meaning. *ACM SIGMOD Record*, 23(4):34–41, December 1994. (refereed as the invited paper for special issue on metadata for digital media).
- [11] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.
- [12] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Cornell University, Ithaca, NY, USA, 1987.
- [13] Stedmans. *Stedman's Cardiovascular & Pulmonary Words: Electronic Word Book Includes Respiratory*. Lippincott Williams & Wilkins, 4th cd-rom edition edition, May 2004.

The Architecture of an Intelligent Agent in MAS

Nikola CIPRICH, Marie DUŽÍ, Tomáš FRYDRYCH, Ondřej KOHUT,
Michal KOŠINÁR

*Department of Computer Science, FEI, VŠB – Technical University Ostrava,
17. listopadu 15/2172, 708 33, Ostrava - Poruba, Czech Republic*

Abstract. In the paper we propose a Brain Architecture which allows developers of a Multi-Agent System (MAS) to integrate various supporting tools. For the purpose of agents' communication and reasoning we make use of *Transparent Intensional logic (TIL)*, or more exactly of its software variant the *TIL-Script language*. The proposed architecture makes it possible to utilise also the *Prolog language* as a reasoning tool. Rules and facts are stored in agent's internal knowledge base is designed in a way appropriate both for Prolog and TIL-Script languages. The architecture is an open one so that other tools of reasoning can be easily incorporated.

Keywords. Multi-agent system, brain architecture, reasoning, Transparent Intensional Logic, TIL-Script language, Prolog

Introduction

Artificial Intelligence and large computer systems use a vast range of multi-agent applications.¹ Multi-agent system is a system of autonomous, intelligent, but resource-bounded agents. Agents communicate with each other, and make decisions on their own in order to achieve their individual as well as collective goals. To this end they are equipped with a 'brain'. In the paper we propose a universal architecture of agent's brain which makes it possible to utilise various tools of reasoning.

The paper is organized as follows. In Section 1 we propose the architecture of agent's brain. Section 2 introduces *Brain Interface* which deals with message inputting and outputting. In the Section 3 we discuss the usage of *Prolog Inference Unit*. Section 4 deals with the *TIL-Script* language and its usage in a multi-agent system. In Section 5 we describe some support modules, namely the *Internal Knowledge Base* and the *Auxiliary Computational Modules*. Section 6 describes the communication of agents in MAS, and finally in Section 7 we propose the method of knowledge acquisition from *Geographical information systems*.

¹ The basic framework for multi-agent approach can be found in [9].

1. Brain Schema

The schema of the *brain architecture* is illustrated by Figure 1. The architecture is a modular and open one. If needed, other units can be easily incorporated. The first proposal of the *brain architecture* was presented in [1]; here we introduce a more detailed proposal of the architecture and describe its important units.

In order to make decisions, the brain must contain an *inference unit*. Our architecture supports the usage of a number of inference units which can be based on various technologies. Currently we develop two inference units, namely *Prolog Inference Unit* and *TIL Inference Unit*. The former makes use of the first-order logic programming in *Prolog* and has been already tested on the data of a traffic system. The latter is still work in progress.

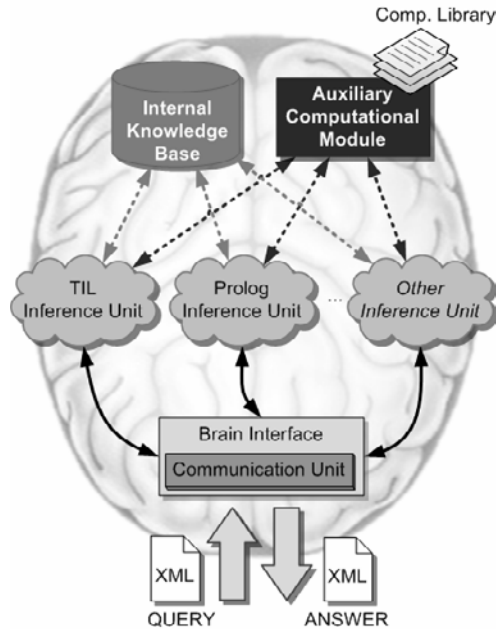


Figure 1. Schema of a brain

In order to function in an environment, an agent must be able to remember some facts and learn new pieces of knowledge. Thus agent's brain has a memory component, namely the *Internal Knowledge Base*. All inference units have an access to services provided by the *Auxiliary Computational Module*, the component that performs sundry helpful mathematical computing (graph algorithms, numeric method, etc.). The brain is interconnected with the "body" via the *Brain Interface* that serves for message inputting and outputting, their decoding and assigning to particular inference units.

2. Brain Interface

If we want to design an intelligent system, then its agents have to be able to make correct deductions. This facility is realised by means of the *inference units* which are

connected to the *Brain Interface*. The brain may contain a number of inference units, and each unit can be based on a different logical mechanism.

Agent's brain communicates with its 'body' (i.e., process units capable of executing particular actions) *via* messages encoded in the XML format. When the *Brain Interface (BI)* receives an XML message, it decides which brain unit is a proper one to deal with the message. Messages coming from other agents are assigned to the *Communication Unit*, which is a part of *BI*. Other messages are directly forwarded to a selected Inference Unit. The respective inference unit performs relevant deductions, and the resulting answer is returned to the *Brain Interface*. The choice of a suitable inference unit is realised by applying a pre-defined map $f: P \rightarrow U$, where P corresponds to a set of problems and U is the set of inference units. The XML message pack contains also the unique name of a problem. If the problem does not belong to the domain of the map f then the agent asks other agents for help (by means of the *Communication Unit*); if no help is available then the agent simply executes a default action.

Communication with other agents is dealt with by the *Communication Unit*. The content of a message is encoded in the *TIL-script* language which has been chosen as a basic communication tool of the system and will be described in Section 4. The input message is decoded by the Communication unit. If it is possible to handle the content by Prolog means, it is transformed into Prolog language, and forwarded to Prolog Unit. Otherwise the TIL-Script Unit is assigned to deal with the message content.

3. Prolog Inference Unit

The *Prolog Inference Unit* has been implemented and tested using the traffic case study data. The unit consists of two main parts: the *PIU interface* and the *Prolog Engine*.² The former communicates with the *Internal Knowledge Base*, and the latter infers consequences from the known facts using *Prolog* rules. The schema of the *Prolog Inference Unit* is shown in Figure 2.

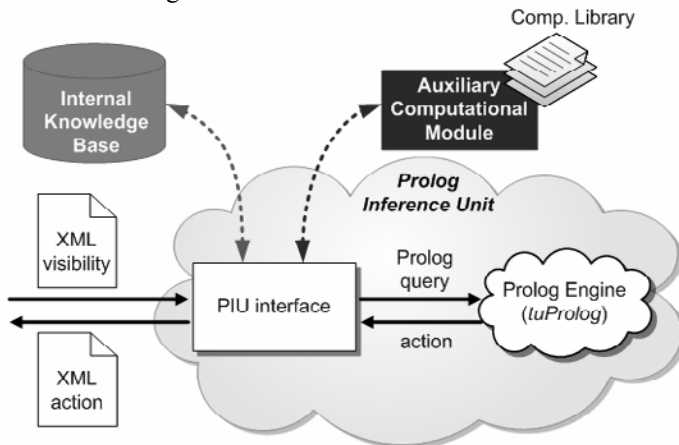


Figure2. The architecture of the tuProlog Engine.

² We use *tuProlog* open source implementation written in *Java* Language; see <http://www.alice.unibo.it:8080/tuProlog/>

PIU Interface transforms incoming ‘visibility’ (i.e., the description of the environment within the area visible to a particular agent) encoded in the XML format to a *Prolog query* and updates *Prolog rules* in accordance with the *Internal Knowledge Base*. Then the *Prolog Engine* is called for to solve the generated query. The inference unit can also use the services of the *Auxiliary Computational Module* if a complicated calculation is needed. When an answer is deduced from Prolog facts, it is packed into the XML format and returned to the *Brain Interface*. If no solution is found then the *Prolog IU* informs the *Brain Interface* and the control is switched to the *Communication unit*.

4. TIL Inference Unit

As mentioned above, the *TIL-Script* language has been chosen as a communication and specification language of our MAS system.³ The *TIL-Script* integration into the *Brain architecture* follows the general architecture schema, the Prolog instance of which has been described in the previous Section 3. Thus the roles of particular components are similar to those of the Prolog inference unit. The *TIL IU Interface* deals with the communication with the *Internal Knowledge Base*, *Auxiliary Computational Modules* (if needed), and in particular with the *Brain Interface*.

The *TIL-Script language* is a software variant of *Transparent Intensional Logic (TIL)*. TIL is a logical system founded by Pavel Tichý.⁴ It is a higher-order system primarily designed for the logical analysis of natural language. As an expressive semantic tool it has a great potential to be utilised in artificial intelligence, and in general whenever and wherever the humans need to communicate with the computers.⁵

The key notion of *TIL* is that of a *construction*. It is an algorithmically structured procedure, or instruction on how to arrive at a less structured entity, the product of the procedure. Constructions are assigned to expressions as their meanings. From the formal point of view, TIL is a partial, typed λ -calculus. However, whereas terms of λ -calculi are mere sequences of symbols that have to be interpreted in order to equip them with meaning, TIL λ -terms denote (encode) directly *constructions*. Moreover, the meaning of the respective λ -term is just the *construction* of a denoted function rather than the function itself.

TIL is well-suited for the utilization as a content language in multi-agent systems. Its main advantages are *compositional procedural semantics* and *high expressibility*. Thus the communication between human and computer agents using the *TIL-Script language* can be realised in a smooth way close to human reasoning.⁶

³ For details on the TIL-Script language see Ciprich, Duží, Košinár (2008), in this proceedings, and also [5], [6] and [7].

⁴ For details see [3], [4], [12], [13].

⁵ More on the role of logic in artificial intelligence see [2].

⁶ Our future plans include the inter-connection of the TIL-Script with the semantic web facilities such as WordNets, SynSets and Verb Lexicons.

5. Supporting Modules

The *Internal Knowledge Base* and *Auxiliary Computational Modules* are designed in such a way that the other brain units and Java objects have a direct access to them.

Agent's *Internal knowledge base (KNB)* contains the stable rules and definitions obtained from the shared ontology as well as particular facts and rules that the agent learns during its life-cycle. For the sake of compatibility and portability the *KNB* is implemented in a layer over the SQL compatible relational database management system. Since *KNB* serves as the knowledge exchange system for particular Inference units, which can be realised in different technologies (currently Prolog and TIL-Script), the *KNB* format has to be independent of a particular language. To this end we use linear text entries that are decoded into particular relations as needed.

The general format for storing rules is the form of the following linear entry of relations:

Rule (head, arguments, body, timestamp, type)

Fact (head, arguments, timestamp, type),

where *head*, *arguments* and *timestamp* are unique for every entry in the knowledge base (primary key) and *type* is a name of the used language (currently either *Prolog* or *TIL-Script*).

The *TIL-Script* rules and facts are stored in the same form as *Prolog* rules and facts, but the knowledge management is different. In this case the head is optional and if it is not defined the Knowledge base auto-generates one. The distinction between rules and facts in *TIL-Script* and those of *Prolog* is given by the form of TIL constructions. If the lambda Closure abstracts from world and time variables only, then the whole sentence is a fact, as illustrated by the following example:

```
Kazuya, Jin/Indiv.
Father/(Bool Indiv Indiv)@Time World.
\w\t['Father@wt 'Kazuya 'Jin].
```

Gloss: the first two constructions introduce TIL types of particular entities: *Kazuya* and *Jin* are individuals, and *Father* is a relation-in-intension (parameters *Time* and *World*) between individuals. The third construction introduces the fact that *Kazuya* is a father of *Jin*. (TIL-Script notation for Lambda abstraction is '\'.)

A simple *TIL-Script* rule specifies the relation-in-intension of a grandfather:

```
Grandpa := \w \t \x, y [Exist z [Impl [And ['Father@wt x z]
[Father@wt z y]] ['Grandpa@wt x y]]].
```

The suffix '@wt' (application to the world and time variables) is interpreted by the Inference unit as the signal to search the knowledge base and check whether there is a fact corresponding to this rule in order to return Yes or No (or undefined if an exception occurs due to partiality) as an answer to a particular query.

Figure 3 illustrates a simple communication between agents A and B using the above facts and rules as processed by Prolog engine. The new facts obtained by knowledge exchange are underlined. All the newly learned facts and rules are stored in the *Internal Knowledge Base* with the actual system timestamp.

Knowledge base is currently used for agents' learning by knowledge exchange. In future we plan to use the Knowledge Base also for active teaching. The design and implementation of a full-fledged knowledge management (including forgetting non-actual and irrelevant knowledge) is also a subject of future research.⁷

⁷ For more details on knowledge management using *Transparent Intensional Logic* see also [8].

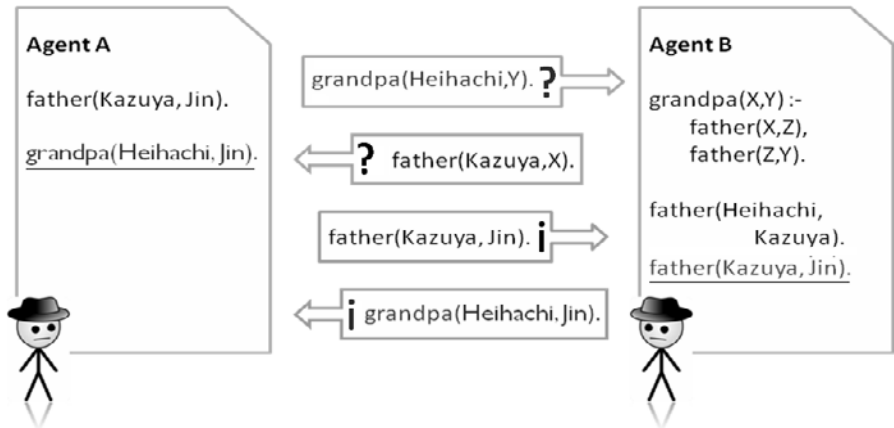


Figure 3. Schema of the Prolog knowledge exchange

Particular inference units are designed to solve different types of problems. Since it is not efficient to implement the same algorithms and computational methods in various units, they are concentrated in the *computational module (CM)*. Moreover, particular logical tools are usually not suited well to execute complex numeric computations. Thus *CM* serves as a computational base of built-in algorithms and a computational library.⁸

6. Communication in MAS

The Foundation for Intelligent Physical Agents (FIPA)⁹ introduced basic standards for MAS (see [10], [11]). According to FIPA standards, an ACL *message* is the basic unit of communication. A message can be of an arbitrary form but it should contain a set of attributes. The most important attributes of communication are ‘Performative’, ‘Content’ and ‘Ontology’.

The crucial role among these attributes is played by the *content* of a message. Performative denotes the type of a message. Basic performatives are Query, Inform and Request. The semantic *content* of a message can be encoded in any suitable language. Ontology is a vocabulary of the domain-specific terms. These (and only these) terms can be used in a content of a message.

Example. Now we demonstrate the communication of two agents, *A* and *B*, by exchanging ACL messages. The content of messages is encoded in the TIL-Script language. The agent *A* is a ‘dummy agent’ who can detect an obstacle but does not know how to overcome it. Thus *A* needs to find agents who can deal with the obstacle (in this case *B*). The basic schema of such communication is illustrated by Figure 4:

⁸ Like the other parts of the brain, *CM* is also implemented as a Java object; its methods can be directly used by means of Java calls. For an extension of this library we intend to include an interpret-programming language for a runtime extension without the need of compilation.

⁹ See <http://fipa.org/>

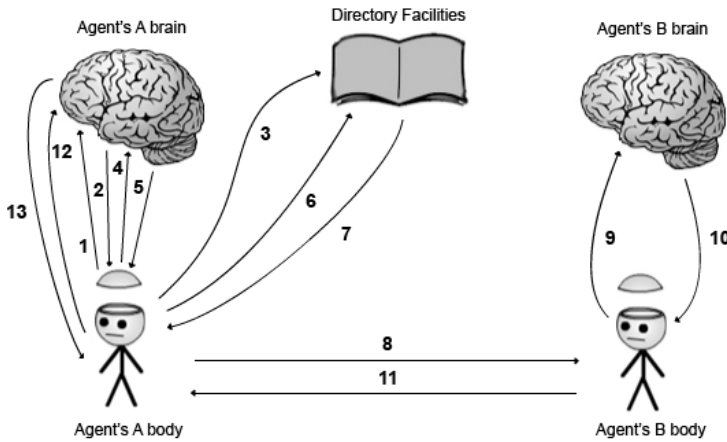


Figure 4. Schema of agent's communication.

Particular steps of the decision making process (after the agents are 'born') are the following:

1. Agent's *A* body asks the brain for a minimum knowledge set.
2. The brain returns the minimum knowledge set to the body.
3. *A* inserts its basic knowledge into the yellow pages (*Directory Facilities*).
4. *A*'s brain receives a message on visibility, which indicates an obstacle.
5. *A*'s brain sends a message to its body that it does not know how to deal with the obstacle.
6. *A* sends a query to *Directory Facilities* (*DF*) searching the agents who can deal with the obstacle.
7. *DF* returns the list of agents capable to deal with the obstacle to the agent *A* (in this case the list contains only the agent *B*).
8. *A* sends an ACL message to *B*, the content of which is the query (encoded in *TIL-Script*) on how to deal with the obstacle.
9. *B* receives the message and consults its *Internal Knowledge Base*.
10. *B*'s brain acquires the necessary piece of knowledge from the *Internal Knowledge Base* and returns it to the body.
11. *B* (its body) sends an ACL message to *A* containing the answer.
12. *A* receives the message and stores its content into the *Internal Knowledge Base*.
13. *A* makes a decision and sends the result to its body asking to execute the respective action in order to overcome the detected obstacle.

In reality, the situation is not so simple, of course. In agents' mutual communication two situations may occur: agents send *rules*, or *methods*. The rules applied to particular arguments yield logical actions, which is dealt with by the brain. Each rule has to correspond to methods which are implemented in the body. The methods are instructions how to execute an action with given parameters on the physical level. These actions are body's task, and the brain does not have to know their detailed description. All the rules and corresponding methods are stored in the Internal Knowledge Base.

Each *method* consists of the so-called *atomic actions*. Atomic action is an action that the body is capable to execute without the interference of the brain. Each atomic action is identified by its name (the key-word contained in the ontology), and it has assigned its implementation code. When sending a method, the agent receives the sequence of atomic actions which are to be executed, and their execution is controlled by the brain.

7. Knowledge acquisition from GIS

Agents have to be able to take into account spatial aspects of their environment. *Geographic information systems (GIS)* are used for gathering, analysis and visualization of information on the space aspects of real-world objects. The main advantage of *GIS* is the ability to relate different kind of information obtained from different sources of a spatial context. This ability enables the agents to act in the real-world environment and make decisions based on its state. *GIS* ontology makes it possible to receive, manipulate and exchange spatial information. In general, *GIS agents* supply the other agents with their ‘visibility’, i.e., the description of the space within the area visible to a particular agent. For more details see [6].

Spatial agents make use of geographic information as the input source for the decision-making mechanism. *Situated agents* are context-aware. They are aware of their position in space, actual speed, surrounding objects and relationships between them. An agent can perceive its environment by sensors or by using the geographic database.

8. Conclusion

In this paper we introduced the architecture of an intelligent agent in MAS. We concentrated on agents’ brain architecture and the mechanism of decision making. Our aim is to develop the brain as universal as possible. Thus in search for an optimum solution we combine several technologies. The architecture is designed as an open one; the entire brain is implemented in Java language, and the units make use of Prolog, TIL-script, Python and SQL relational database management tools. Due to its modular architecture the brain design is very flexible. We can add other inference units using other technologies, if needed, to solve some special tasks. When a new communication language is needed then only the Communication Unit has to be updated. Thus the system can also be easily adapted as a *multi-lingual system*. Due to a shared ontology and the expressive power of the TIL-Script language agents’ communication can be smoothly realized in various languages.

The existing communication standards for multi-agent systems are syntactically rather than semantically defined. This can slow down the development progress. Therefore we proposed the TIL-Script language which is based on the well-elaborated Transparent Intensional Logic. TIL-Script is a semantically driven language suitable for MAS communication. Its high expressive power makes it an appropriate tool to adopt other logics and languages into its semantic framework, so that TIL-Script can be used as a general specification language. The TIL-Script semantic facilities make it possible to design the communication between humans and computer agents in a smooth and natural way.

The TIL-Script language is being implemented and tested using the Python language and the framework Jadex¹⁰. The SQL knowledge base database management system is implemented in Open Source SQLite, and for the sake of maximum portability (Sun Java and .NET support) its control layer is implemented in the Python language.

Acknowledgements. This research has been supported by the program ‘Information Society’ of the Czech Academy of Sciences, project No. 1ET101940420 “Logic and Artificial Intelligence for multi-agent systems”.

References

1. Kohut, O.: Brain for agents in multi-agent systems, In *WOFEX 2007*, Faculty of electrical engineering and computer science, VŠB – Technical University of Ostrava, 2007, p. 291-296
2. Thomason, R.: *Logic and Artificial Intelligence*[online]. The Stanford Encyclopedia of Philosophy, Available from WWW: <<http://plato.stanford.edu/archives/sum2005/entries/logic-ai/>>.
3. Duží, M., Jespersen, B., Müller, J.: Epistemic Closure and Inferable Knowledge. In *the Logica Yearbook 2004*. Ed. Libor Běhounek, Marta Bílková, Praha:Filosofia, 2005, Vol. 2004, 124-140, Filosofický ústav AV ČR, Praha, ISBN 80-7007-208-3.
4. Duží, M., Materna, P.: *Constructions*, <<http://www.phil.muni.cz/fil/logika/til/>>.
5. Ciprich N., Duží M., Košinár M.: Functional Programming Based on Transparent Intensional Logic. In *RASLAN 2007*, Masaryk University Brno, pp. 37-42
6. Kohut, O., Košinár, M., Takács, O.: Brain Architecture and Reasoning of Intelligent Agents in MAS, In *GIS Ostrava 2008*, Faculty of electrical engineering and computer science, VŠB - Technical University of Ostrava, 2008
7. TIL-Script Home Page [online]. c2007 [cit. 2008-1-10]. Available from WWW: <<http://www.cs.vsb.cz/til/>>.
8. Gardoň A., Horák A.: The Learning and Question Answering Modes in the Dolphin System for the Transparent Intensional Logic. In *RASLAN 2007*, Masaryk University Brno, pp. 29-36
9. Luck, M., McBurney, P., Shehory, O., Willmott, S.: *Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing*. University of Southampton on behalf of AgentLink III, 2005.
10. FIPA: *FIPA SL Content Language Specification* [online]. c2002 [cit. 2007-11-11]. Available from WWW: <<http://www.fipa.org/specs/fipa00008/>>.
11. FIPA: *FIPA Abstract Architecture Specification* [online]. c2002 [cit. 2007-11-11]. Available from WWW: <<http://www.fipa.org/specs/fipa00008/>>.
12. Tichý, P.: *The Foundations of Frege's Logic*, Berlin, New York: De Gruyter 1988.
13. Tichý, P.: *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press, 2004

¹⁰ <http://vsiis-www.informatik.uni-hamburg.de/projects/jadex/>

A Meta-Level Knowledge Base System for Discovering Personal Career Opportunities by Connecting and Analyzing Occupational and Educational Databases

Yusuke Takahashi[†] and Yasushi Kiyoki[‡]

[†] Graduate School of Media and Governance, Keio University

Endo 5322, Fujisawa, Kanagawa, #252-8520, JAPAN, yt@sfc.keio.ac.jp

[‡] Faculty of Environment and Information Studies, Keio University

Endo 5322, Fujisawa, Kanagawa, #252-8520, JAPAN, kiyoki@sfc.keio.ac.jp

Abstract. This paper presents an implementation method of a meta-level knowledge base system for analyzing personal career opportunities by connecting occupational and educational databases. We have designed and implemented several functions to analyze information on personal career development, on the meta-level database system for connecting occupational and educational databases. This method is used to create dynamic relationships among heterogeneous databases on occupations, educational contents, social and educational issues and personal career information. In this method, several functions are defined for analyzing relationships among heterogeneous databases. (1) Educational contents represent, for instance, lectures in educational institutes, (2) occupations represent descriptions of concrete occupations, (3) social and educational issues represent academic industrial fields, and (4) personal career information represents histories, objectives and interests of an individual user's career. These databases are connected and analyzed in a dynamic way, according to users' contexts and situations. By using our method, individual career development becomes to be effectively supported when discovering personal career opportunities and designing personalized career development plans.

1 Introduction

In a multidatabase environment, it is important to realize the connection merit between heterogeneous domain fields, and besides, it is also important to introduce experts' knowledge for evaluating those heterogeneous information for creating new effective knowledge base applications.

1.1 Background Issues

Supporting personal career development in a effective way is an important issue in societies and multidatabase and knowledge base technologies are essentially significant for providing various opportunities to extract personal abilities. In our own career development, we observe our current skills, experiences, significant knowledge and career goals, and consider our targeted occupations. We need to know what we should learn for our career development, what our career image in our future is, and what can expand our opportunities for our future's career.

However, it is difficult to collect and obtain related and important information for our personal career development by ourselves, because there are various kinds of information resources which are almost infinite. There exist numerous kinds of candidates of careers in both domestic and international employment markets, and they are provided in every single moment. On the other hand, personal career objectives and interests vary depending on each person. Furthermore, these kinds of personal career information, including histories,

objectives and interests, are updated as they develop their career. It also changes in a dynamic way, according to the changes of social situations.

1.2 Solutions

This paper presents an implementation method of a meta-level knowledge base system for analyzing personal career opportunities by connecting occupational and educational databases.

We have proposed several functions to analyze information on personal career development, on the meta-level database system for connecting occupational and educational databases, that we have presented in [1, 2, 3]. This system is described as the basic system architecture and data structures. This paper focuses on the implementation of this system for actual applications. This implementation method is used to create dynamic relationship among heterogeneous databases on occupations, educational contents, social and educational issues and personal career information.

In this method, several functions are defined for analyzing relationships among heterogeneous databases. Educational contents represent, for instance, lectures in educational institutes, occupations represent descriptions of concrete occupations, social and educational issues represent academic and industrial fields, and personal career information represents histories, objectives and interests of individual user's career. These databases are connected and analyzed in a dynamic way according to users' contexts and situations.

This system realizes numerical computation and quantitative analysis among heterogeneous databases on occupations and educational contents, that have been difficult with the previous approaches where experts of different fields separately and manually collect information and evaluate career information depending on knowledge and experiences of their own fields.

1.3 Related Works

The essence of our system is described as following points, which are (1) interoperability and (2) dynamic evaluation using experts' knowledge and users' contexts.

Recently, several kinds of services are open to the public [4, 5] and various academic research are committed [6, 7]. Although these are respectively good services and fine results, they are separated and do not have interoperability for calculating relationship between occupations and educational contents. Our system enables interconnection and interoperability for calculating such heterogeneous information, that has strong relation in a sense of a career development.

For supporting personal career development, given information should cover a wide range of knowledge such as vocational descriptions and requisites, effects of education and training, and it should be related and organized in a dynamic way according to the users' contexts, their intension and expert's knowledge on career development.

By using our method, individual career development becomes to be effectively supported when discovering personal career opportunities and designing personalized career development plans.

1.4 Applications to Actual Career Development

Our system focuses on a period of "stop to design your career at *crossroads*" in a *Transition Cycle Model* (Figure 1) [6]. At this period of this model, users design their careers. After this, they take action, drifts their career and again, at *crossroads* of their career, they stop to think of their careers. The main users of our system are classified as follows:

- **Students in Colleges and Schools**, who select majors; who design their careers.
- **Workers**, who design their careers.

Our system is applicable on a wide scale, especially to:

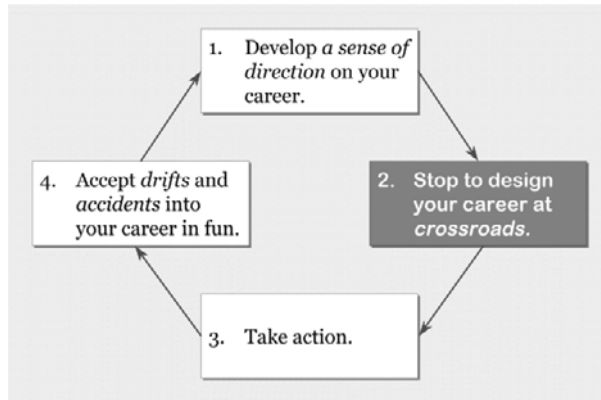


Figure 1: *Transition Cycle Model* in terms of Career Design [6]: Our System Focuses on Users at the Period of Block-2. *Stop to design your career at crossroads.*

- **Designing Personalized Curricula or Programs in Colleges and Schools.** In colleges, students have to develop their curricula by selecting lectures related to their courses. They decide which classes to take during not only the periods of liberal arts but also the whole periods before completing credits. Variety of students enters into professional schools. Their varied background can not fit to legacy static curricula.
- **Indicating Candidates; Path of Career.** At any period our life, we have to stop to think of and design our career.

In the following, we describe general existing processes of personal career development to show what our system realizes to expand limitations of current situations. We review the basic system architecture of our system, and then show implementation of a meta-level knowledge base system for discovering personal career opportunities. We perform several experiments to show its feasibility, and finally, we discuss applicability and future works of our system.

2 Processes of Personal Career Development

In this section, we describe general existing processes of personal career development to show what our system realizes to solve problems and expand limitations on it.

2.1 Examples of Personal Career Development

This paper presents a meta-level knowledge base system for discovering personal career opportunities by connecting and analyzing occupational and educational databases. This system is, in other words, a dynamic computing and analyzing system, that connects users' career information such as histories, objectives and interests and related information resources including occupational descriptions and educational contents, on a meta-level system. It evaluates relationship among those information depending on users' situation and context, by using experts' knowledge on career development and curriculum design.

For demonstrating what our system realizes, we define career descriptions of virtual users, User-A: **Andrea** and User-B: **Barry**, in colleges or schools (**Phase-1**) and at their career changes (**Phase-2**). The stories of the examples are described as follows.

Phase-1 and **Phase-2** is corresponding to the period of "Stop to design your career at crossroads" in a *Transition Cycle Model* (Figure 1) [6].

Table 1: Career descriptions of virtual users, User-A: **Andrea** and User-B: **Barry**, in colleges or schools (**Phase-1**) and at their career changes (**Phase-2**), current situations, and advantages in the functions of our system. KSAs is an abbreviation of Knowledge, Skills and Abilities.

	In College and Schools (Phase-1) and At Career Change (Phase-2)
User-A: Andrea	<p>Phase-1: majored in <i>Graphic Design</i>.</p> <p>Phase-2: started his own business and active working as a <i>Graphic Designer</i>, a <i>Fashion Designer</i> and a <i>Multi-Media Artist</i>.</p>
User-B: Barry	<p>Phase-1: majored in <i>Database Technologies (Computers and Electronics)</i>.</p> <p>Phase-2: working as a <i>Database Administrator</i> and a <i>Computer Programmer</i> in a company.</p>
Current Situations	<ul style="list-style-type: none"> • Majors and lectures are chosen by preferences, without knowing every possibility available. • Introduction of career opportunities by their guiding professors and career development department of their college are helpful. Articles on target career fields and comments by role models on mass media are also referable, if we ignore comprehensive knowledge of career opportunities, educational contents, effects of training and inter-relations among these. • Job seeking services are helpful for just looking for career opportunities, if we ignore relationship among requisites for the job, educational contents and personal career information. • Intuitive selection or accidental discovery of career opportunities guides you succeed, once in a while.
Advantages in the Functions of Our System	<p>I. Discover career candidates that are appropriate to your career histories.</p> <p>II. Discover career opportunities assuming that additional KSAs are acquired.</p> <p>III. Discover <i>crossover</i> career opportunities over the various types of industries.</p> <p>IV. Discover career alternatives to your career goals in terms of KSAs.</p> <p>V. Design curriculums for acquiring important KSAs to realize career goals.</p> <p>VI. Design curriculums for acquiring missing KSAs to realize career goals.</p>

At **Phase-1**, in college, **Andrea** majored in *Graphic Design*, and **Barry** majored in *Database Technologies (Computers and Electronics)*. Both of them chose their majors because they preferred them.

At **Phase-2**, several years after their graduation, **Andrea**, after having started her own business, has been active working as a *Graphic Designer*, a *Fashion Designer* and a *Multi-Media Artist*, and **Barry**, in a company, as a *Database Administrator* and a *Computer Programmer*. When graduating from college, they selected their occupations out of information they have found by introduction of their guiding professors and career development department of their college. Some job seeking services [4, 5] were also helpful for looking for career opportunities. They have read many articles on their target fields and referred to comments of their role models on mass media.

2.2 Current Situations of Personal Career Development

Existing information available for career development at both phases in college or school (**Phase-1**) and at their career change (**Phase-2**) are limited in the sense of coverage of information, reference to related information and special knowledge for judgement

In many cases, majors and lectures are chosen by preferences, without knowing every possibility available. Introduction of career opportunities by their guiding professors and career development department of their college are helpful. Articles on target career fields and comments by role models on mass media are also referable, if we ignore comprehensive knowledge of career opportunities, educational contents, effects of training and inter-relations

among these. Job seeking services are helpful for just looking for career opportunities, if we ignore relationship among requisites for the job, educational contents and personal career information. Intuitive selection or accidental discovery of career opportunities guide you succeed, however such cases happen once in a while.

2.3 Advantages in the Functions of Our System

Our system has following advantages.

- I. Discover career candidates that are appropriate to your career histories.
- II. Discover career opportunities assuming that additional KSAs are acquired.
- III. Discover *crossover* career opportunities over the various types of industries.
- IV. Discover career alternatives to your career goals in terms of KSAs.
- V. Design curriculums for acquiring important KSAs to realize career goals.
- VI. Design curriculums for acquiring missing KSAs to realize career goals.

These advantages are beyond human abilities, and they are essentially important for helping people discover career opportunities. We have realized them as a meta-level functions on a knowledge base system in a multidatabase environment. These functions realize following superhuman abilities.

- Comprehensive Discoverability over Heterogeneous Information Resources
- Interoperability between Heterogeneous Information Resources
- Advanced Knowledge and Technologies for Application

The point is that these functions are realized only with simple and easy-to-implement functions on a meta-level system.

With previous approaches for supporting career development, we have to try hard to find all the information in charge of our personal career development. We are required to integrate, edit and evaluate by ourselves manually by introducing our knowledge and experiences limited in the sense of coverage of information, reference to related information and special knowledge. There exist no superhuman experts who has comprehensive knowledge of career development including educational and occupational information and interrelation between them in the sense of career development.

Our system support decision-making effectively for both in college or schools when users have concrete career goals and want to know what he should do to realize them, and furthermore, when users have no concrete career goals and hope to gather information that expand his career possibilities.

Career descriptions of virtual users, **Andrea** and **Barry**, in colleges or schools (**Phase-1**) and at their career changes (**Phase-2**), current situations of personal career development, and advantages in the functions of our system are described in the Table 1.

3 A Meta-Level Database System Connecting Occupational and Educational Databases

In this section, we review a method to design and construct a meta-level database system connecting occupational and educational databases [1, 2, 3], that describes the basic system architecture and data structures for a meta-level knowledge base system for analyzing and discovering personal career opportunities. First, we describe our basic concept, and then, explain three components that construct a meta-level database system connecting occupational and educational databases.

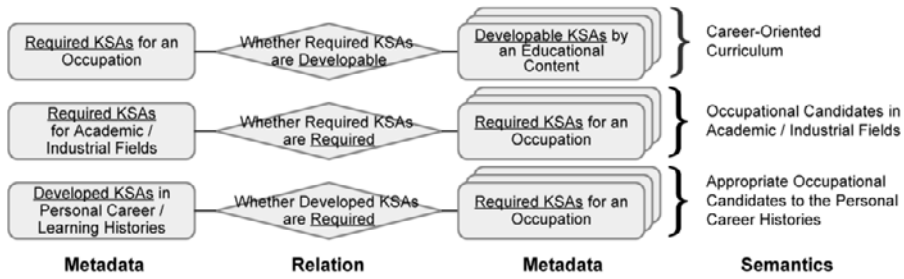


Figure 2: The Basic Concept : We evaluate of interrelation between heterogeneous databases on, such as occupations, educational contents and personal career information, regardless of difference of data representation, and according to inherent semantics of content and external conditions of the application and users' contexts.

3.1 The Basic Concept

Our method has following two basic features for connecting heterogeneous databases that are related to career development.

- An ability to evaluate interrelation between heterogeneous databases on, such as occupations, educational contents and personal career information, **regardless of difference of data representation**. We introduce common features for expressing respective databases as metadata for the application of career development.
- An ability to evaluate interrelation between heterogeneous databases on, such as occupations, educational contents and personal career information, **according to inherent semantics of content and external conditions of the application and users' contexts**. We generate metadata with particular meanings for applications respectively, define particular meanings on interrelation between those metadata for users' context respectively, and then, a set of results of calculating correlation is given particular meaning.

For example, there exists a semantic relationship of whether required KSAs for an Occupation are Developable by an Educational Content. Then the set of educational content, such as lectures, where **Required KSAs** for an Occupation are **Developable**, corresponds to of **Career-Oriented Curriculum for the Occupation**. In the same way, there exists a semantic relationship of whether Developed KSAs are Required in Personal Learning Histories (Career Histories) for an Occupation. Then the set of occupations, where **Developed KSAs** in Personal Career Histories are **Required**, corresponds to **Occupational Candidates Appropriate to the Personal Career Histories** (Figure 2).

We explain three components, as listed below, that construct a meta-level database system connecting occupational and educational databases in the following parts (Numbers (1) - (3) are corresponding to those shown in the Figure 3).

1. Semantic space creation with technical terms of career development for semantic vector computation
2. Special metadata generation for career development
3. Semantic relation evaluation corresponding to various objectives of career development

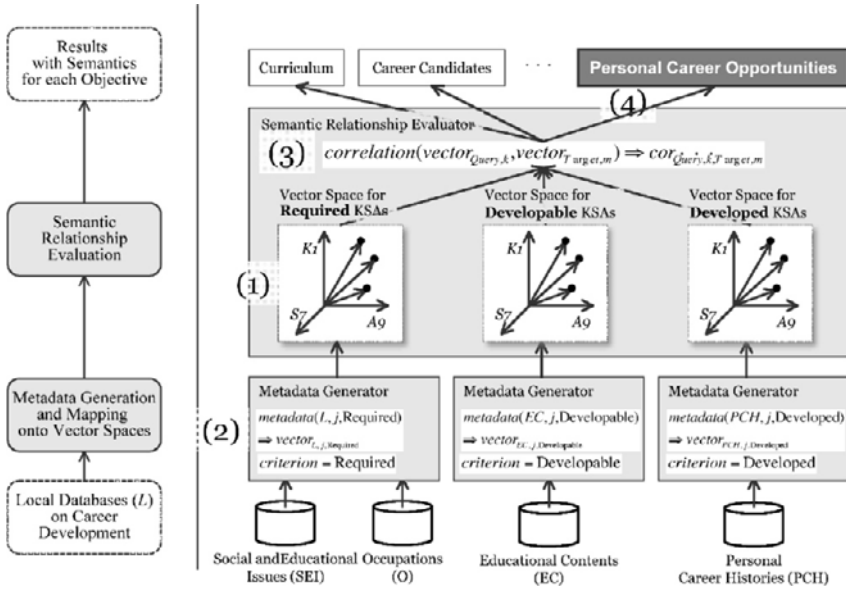


Figure 3: Implementation Method : We evaluate each data item of Occupations(*O*), Educational Contents(*EC*), Social and Educational Issues(*SEI*) and Personal Career Histories(*PCH*) in the same metric to map them on a vector space, and then measure correlations. (1) - (3) in the figure are our previous system, which are explained in the Sections 3.2, 3.3, 3.4, and (4) is the method we propose in this paper, which is explained in the Section 4.

3.2 Semantic Space Creation with Technical Terms of Career Development for Semantic Vector Computation

Our method integrates and connects heterogeneous databases related to personal career development, and therefore it is important for our method to discuss following features:

- **System Environment** for connecting heterogeneous databases concerning personal career development.
- **Computational Model** for evaluating relationships among heterogeneous databases on educational issues, occupations, educational contents and personal career histories.
- **Metrics** for evaluating each databases, such as specialist's knowledge in a field of personal career development.

3.2.1 System Environment

Our system connects heterogeneous databases on personal career development, and evaluate each data item comprehensively. Therefore, our system has to meet the conditions such as:

- Legacy databases, that are constructed separately, should be integrated by using expert's knowledge on personal career development.
- Connected databases to our system, such as those on occupations, educational contents, social and educational issues and personal career histories, should be constructed and authorized separately, because these fields are mutually irrelevant.
- Continuous updates of each database should be flexibly managed.

Our system is implemented as a **multidatabase system** [8, 9, 10]. A multidatabase system is used for unifying independently-built databases to share them like a single database. A multidatabase system realizes interoperability among existing databases and adds newly applied use and new values to existing information. The details of its implementation method are shown in [11, 12, 13, 14]. Our system creates a meta-schema in meta-level environment, where the heterogeneity of legacy databases are canceled. It dynamically connects legacy databases including descriptions of social and educational issues, occupations, educational contents and personal career histories in a meta-level system by using expert's knowledge on personal career development. When parts of databases are updated, our system requires only revisions of updated elements of vectors and we do not need to update the whole databases.

3.2.2 Computational Model

Our system is intended to create a meta-level database system for dealing with databases related to personal career development. Therefore the computational model requires a feature such as:

- Discoverability of every possibility for personal career development.

The function of our system is to connect occupational and educational databases. This can be described, in other words, our system plays a role of information retrieval to specify important document for personal career development.

We introduce a **vector space model** [15, 16, 17] as a computational model for calculating correlation. A vector space model is a model for information retrieval that calculate similarity between a query and a document by expressing queries and documents in a united framework of n -dimensional vectors.

By introducing a vector space model, we extract set of feature words, that is explained as metrics in the following part 3.2.3. Metrics, and values for metadata represents each document. Without defining all the relationships among every document, we can calculate similarity among each document to discover important document for users' personal career development.

3.2.3 Metrics

Our system has functionalities for connecting educational and occupational databases. When evaluating relevance among heterogeneous databases, it is important to correctly extract values of metadata, that represent legacy databases. Our system is intended to create a database system for dealing with databases related to personal career development. Therefore the metric requires features such as:

- Knowledge of the inter-occupational relationships and whole structure of industries, in addition to that of individual occupations.
- Knowledge of requisites, such as professional knowledge, skills, abilities and experiences, that are required for all the individual occupations.
- Information of acquirable knowledge, skills and abilities for individual educational contents.
- Knowledge for evaluating information on personal career development; knowledge of evaluating relevance among educational, occupational and personal career information.

There exist several frameworks that describe set of skills and knowledge and curricula for a specific domain, such as the Project Management Body of Knowledge (PMBOK) [18], the Software Engineering Body of Knowledge (SWEBOK) [19], and the Computing Curricula [20]. Legacy databases for educational contents have descriptions of syllabuses. Occupational databases hold vocational descriptions such as required knowledge, skills and abilities.



Figure 4: O*NETTM is a trademark of the U.S. Department of Labor, Employment and Training Administration.

Table 2: 120 kinds of KSAs from the O*NETTM Content ModelTM including 33 kinds of Knowledge, 35 Skills and 52 Abilities. Refer to the O*NETTM website for detail[21].

Knowledge (K)
Administration and Management / Biology / Building and Construction / Chemistry / Clerical / Communications and Media / Computers and Electronics / Customer and Personal Service / Design / Economics and Accounting / Education and Training / Engineering and Technology / English Language / Fine Arts / Food Production / Foreign Language / Geography / History and Archeology / Law, Government and Jurisprudence / Mathematics / Mechanical / Medicine and Dentistry / Personnel and Human Resources / Philosophy and Theology / Physics / Production and Processing / Psychology / Public Safety and Security / Sales and Marketing / Sociology and Anthropology / Telecommunications / Therapy and Counseling / Transportation
Skills (S)
Active Learning / Active Listening / Complex Problem Solving / Coordination / Critical Thinking / Equipment Maintenance / Equipment Selection / Installation / Instructing / Judgment and Decision Making / Learning Strategies / Management of Financial Resources / Management of Material Resources / Management of Personnel Resources / Mathematics / Monitoring / Negotiation / Operation and Control / Operation Monitoring / Operations Analysis / Persuasion / Programming / Quality Control Analysis / Reading Comprehension / Repairing / Science / Service Orientation / Social Perceptiveness / Speaking / Systems Analysis / Systems Evaluation / Technology Design / Time Management / Troubleshooting / Writing
Abilities (A)
Arm-Hand Steadiness / Number Facility / Auditory Attention / Oral Comprehension / Category Flexibility / Oral Expression / Control Precision / Originality / Deductive Reasoning / Perceptual Speed / Depth Perception / Peripheral Vision / Dynamic Flexibility / Problem Sensitivity / Dynamic Strength / Rate Control / Explosive Strength / Reaction Time / Extent Flexibility / Response Orientation / Far Vision / Selective Attention / Finger Dexterity / Sound Localization / Flexibility of Closure / Spatial Orientation / Fluency of Ideas / Speech Clarity / Glare Sensitivity / Speech Recognition / Gross Body Coordination / Speed of Closure / Gross Body Equilibrium / Speed of Limb Movement / Hearing Sensitivity / Stamina / Inductive Reasoning / Static Strength / Information Ordering / Time Sharing / Manual Dexterity / Trunk Strength / Mathematical Reasoning / Visual Color Discrimination / Memorization / Visualization / Multilimb Coordination / Wrist-Finger Speed / Near Vision / Written Comprehension / Night Vision / Written Expression

What is important is the framework and the methods for evaluating occupational databases, educational contents, descriptions of social and educational issues, and career developers’ attributes in the same metric.

Our system introduces *Content Model* defined by the *O*NET*TM (Figure 4) [21]. *O*NET*, the Occupational Information Network, offers a comprehensive database of worker attributes and job characteristics. As a sponsored project of US Department of Labor, Employment and Training Administration (DOL/ETA), it offers a common language for communication across the economy and among work force development efforts.

The *Content Model* described in the Table 3 is the conceptual foundation of *O*NET*. The *Content Model* provides a framework for classifying, organizing, and structuring *O*NET*’s data. It was developed using research on job and organizational analysis. It embodies a view that reflects the character of occupations and people.

We use the *Content Model* of *O*NET* as the metric for evaluating legacy databases to extract metadata, because this information is useful as features for measuring relationships among the databases dealing with information for career development. It provides definitions and concepts for describing worker attributes and workplace requirements including information about knowledge, skills, abilities (KSAs), interests, general work activities (GWAs), and work contexts.

The 120-dimensional vector consists of 120 feature words, with every values set to *boolean values* of {1,0}. *KSAs* ($KSAs \equiv K \cup S \cup A$) includes 120 kinds of *k_{sa}*, as listed in the Table 2, for constructing vector spaces for career development. Though we have defined *boolean values* for each elements of vectors, *numerical values* such as −1 and *scholar values* are also available.

Table 3: Descriptions of O*NET Content Model[21] including KSAs.

Class	Description
Tasks	Specific work activities that can be unique for each occupation.
Knowledge	Principles and facts that apply to a wide range of situations.
Skills	Developed capacities that facilitate learning and performance of activities.
Abilities	Enduring attributes that influence performance.
Work Activities	General types of job behaviors.
Work Context	Physical and social factors that influence the nature of work.
Job Zone	One of five zones based on experience, education, and training requirements.
Interests	Indicate a person's preferences for work environments and outcomes.
Work Values	Global aspects of work that are important to a person's satisfaction.

3.3 Special Metadata Generation for Career Development

We generate metadata in the form of 120-dimensional vector for every single datum $\{data_j\}_{j=1\dots n}$ in Local Databases LDB_L . The metadata generation function $metadata$ is defined as follows.

$$metadata(L, j, criterion) \Rightarrow vector_{L,j,criterion}$$

Where,

$$\begin{aligned} L & : \text{identifier of } LDB \\ j & : \text{identifier of a } data \text{ in } LDB \\ criterion & \in \{\text{Required, Developed, Developable}\} \end{aligned}$$

L is the identifier of Local Databases (LDB).

$$\begin{aligned} LDB_O & : \text{Local Database of Occupation} \\ LDB_{EC} & : \text{Local Database of Educational Contents} \\ LDB_{SEI} & : \text{Local Database of Social and Educational Issues} \\ LDB_{PCH} & : \text{Local Database of Personal Career Histories} \end{aligned}$$

Every datum in Local Databases LDB_L is expressed as $\{data_j\}_{j=1\dots n}$.

Depending on the nature of respective LDB_L in the sense of typical application for career development, we define *criterion* for generating metadata. If a $data_j$ meets the *criterion*, the $metadata_{j,ksa}$ is set to 1, and otherwise to 0. The *criterion* is defined for LDB_L respectively, such as, whether a *ksa* is **Required** for an Occupation (O),

$$criterion_O = \text{Required},$$

whether a *ksa* is **Developable** for by taking an Educational Content (EC),

$$criterion_{EC} = \text{Developable},$$

whether a *ksa* is **Required** for solving a Social and Educational Issue (SEI),

$$criterion_{SEI} = \text{Required},$$

whether a *ksa* is **Developed** in your Personal Career Histories (PCH),

$$criterion_{PCH} = \text{Developed}.$$

As for Personal Career Histories (LDB_{PCH}), we generate metadata with sequential data of temporal information, to deal with dynamic changes of users' career histories and objectives. We define five kinds of *User Description Vectors*, as described in the Table 4, for individual users

By the function of $metadata$, 120-dimensional vector $vector_{L,j}$ is output, which consists of an array of 120 *boolean values* (bv), 1 or 0, as results of evaluation whether the $criterion_L$ meets the $data_j$.

Table 4: Descriptions of the $UDVs$.

ID	Descriptions	Data Class
UDV_{1time}^{User}	Personal Knowledge, Skills, Abilities and Experiences	S_1
UDV_{2time}^{User}	Interested / Additional KSAs	S_2
UDV_{3time}^{User}	Objectives (as Occupational Titles)	S_3
UDV_{4time}^{User}	Required KSAs for Career Objectives	$S_3 \setminus S_1$
UDV_{5time}^{User}	Future's Career Images	$S_1 \cup S_2 \cup S_3$

3.4 Semantic Relation Evaluation Corresponding to Various Objectives of Career Development

The amount of relation is calculated as a value of correlation between vectors. A correlation computation function *correlation* is defined as follows.

$$correlation(vector_{Query,k}, vector_{Target,m}) \Rightarrow cor_{Query,k,Target,m}$$

Where,

$vector_{Query,k}$: query vector
$vector_{Target,m}$: target vector
Query	: identifier of LDB for query vector
Target	: identifier of LDB for target vector
k	: identifier of a <i>data</i> in LDB_{Query}
m	: identifier of a <i>data</i> in LDB_{Target}
cor	: value of correlation

Both of the query keywords and the data for reference are expressed respectively as 120-dimensional vectors that have the same elements. When query keyword (k in LDB_{Query}) and the data for reference (m in LDB_{Target}) are set to as

$$\begin{aligned} vector_{Query,k}^T &= [bv_{k_1} \quad bv_{k_2} \quad bv_{k_3} \quad \dots \quad bv_{k_{120}}], \\ vector_{Target,m}^T &= [bv_{m_1} \quad bv_{m_2} \quad bv_{m_3} \quad \dots \quad bv_{m_{120}}]. \end{aligned}$$

We introduce Cosine Measure \cos as a general method of calculating cor , and it is expressed as following formula.

$$\begin{aligned} \cos(vector_{Query,k}, vector_{Target,m}) &= \frac{vector_{Query,k} \cdot vector_{Target,m}}{\|vector_{Query,k}\| \|vector_{Target,m}\|} \\ &= \frac{\sum_{i=1}^{120} bv_{k_i} \cdot bv_{m_i}}{\sqrt{\sum_{i=1}^{120} bv_{k_i}^2} \sqrt{\sum_{i=1}^{120} bv_{m_i}^2}} \end{aligned}$$

We evaluate cor between $vector_{Query,k}$ and all the $data_m$ in LDB_{Target} , and then order the set of $data_m$ in LDB_{Target} in the order of the values of correlation (cor).

When LDB_{Query} and LDB_{Target} is specified, the relation between *criterion* of LDB_{Query} and that of LDB_{Target} is defined, and semantic relationship between metadata (*Objective*) is fixed.

Examples of semantic relationship between metadata are shown in the following.

- If the Query is *SEI* and the Target is *O*, the *Objective* is **discovering occupational candidates for an Social and Educational Issues (SEI)**. Then the correlation computation function *correlation* is defined as follows.

$$correlation(vector_{SEI,k}, vector_{O,m}) \Rightarrow cor_{SEI,k,O,m}$$

By calculating correlation between Required KSAs for Social and Educational Issues and Required KSAs for an Occupation, this function outputs Occupational Candidates in the Social and Educational Issue as a set of occupations with ordered values of correlation. We obtain, for example, career candidates for curriculums after graduating an educational institute or a concrete descriptions of required talented person for a project.

- If the Query is *SEI* and the Target is *EC*, the *Objective* is **designing curriculum for Social and Educational Issues (SEI)**. Then the correlation computation function *correlation* is defined as follows.

$$\text{correlation}(\text{vector}_{SEI,k}, \text{vector}_{EC,m}) \Rightarrow \text{cor}_{SEI,k,EC,m}$$

By calculating correlation between Required KSAs for Social and Educational Issues and Developable KSAs by an Educational Content, this function outputs Career-Oriented Curriculum for the Social and Educational Issue as a set of lectures with ordered values of correlation. We obtain, for example, a career-oriented curriculum for a concrete social and educational issue or an academic and industrial field.

Our method comes into existence under the condition that the amount of information on each of Occupations, Educational Contents, Social and Educational Issues and Personal Career Histories are all the same. Various contents exist in the real world. We can say that each of Occupations, Educational Contents, Social and Educational Issues and Personal Career Histories has different amounts of information and this kind of information cannot be compared with each other. This is because the amounts of information on existing contents are defined with a specified viewpoint. In other words, metadata on contents is conditioned by a context for each application. We note that our method for measuring correlations works under the condition that each of Occupations, Lectures, Social and Educational Issues and Personal Career Histories has the same amount of information. This is the reason why we normalize vectors before calculating relationships.

4 A Meta-Level Knowledge Base System for Analyzing and Discovering Personal Career Opportunities

In this section, we describe several methods to implement a meta-level knowledge base system for analyzing and discovering personal career opportunities, that is implemented on a meta layer of a meta-level database system connecting occupational and educational databases [1, 2, 3] ((4) in the Figure 3). The method is implemented as following six functions, that are corresponding to 2.3 *Advantages in the Functions of Our System*.

4.1 I. Discover career candidates that are appropriate to your career histories

The function **I** discovers career candidates that are appropriate to user's career histories, and the correlation computation function *correlation* is defined as follows.

$$\text{correlation}(G_{PCH,user}, \text{vector}_{O,m}) \Rightarrow \text{cor}_{G_{PCH,user},O,m}$$

Where,

$$G_{PCH,user} : \text{Gravity Center of Personal Career Histories } \{UDV_1^{user}\}$$

The function **I** requires input of a set of users' Personal Career Histories. Personal Career Histories is expressed as a *User Description Vector* UDV_1 , Personal Knowledge, Skills, Abilities and Experiences. **I** creates UDV_1 as a vector of gravity center $G_{PCH,user}$ and calculate *cor*.

4.2 II. Discover career opportunities assuming that additional KSAs are acquired

The function **II** discovers career opportunities (occupational candidates) assuming that additional KSAs are acquired, and the correlation computation function *correlation* is defined as follows.

$$\text{correlation}(G_{PCH,user} + ksa, \text{vector}_{O,m}) \Rightarrow \text{cor}_{G_{PCH,user}+ksa,O,m}$$

Where,

$$\begin{aligned} G_{PCH,user} &: \text{Gravity Center of Personal Career Histories } \{UDV_1^{user}\} \\ ksa &: \text{Unit vector of an additional } ksa \ (ksa \in \{KSAs\}) \end{aligned}$$

The function **II** requires input of a set of users' Personal Career Histories, an additional ksa . Personal Career Histories is expressed as a *User Description Vector* UDV_1 , Personal Knowledge, Skills, Abilities and Experiences. **II** prepares $G_{PCH,user}$ and ksa , adds them up to create $G_{PCH,user} + ksa$, and then, calculate cor .

4.3 **III**. Discover crossover career opportunities over the various types of industries

The function **III** discovers *crossover* career opportunities over the various types of industries, and the correlation computation function *correlation* is defined as follows.

$$correlation(G_{PCH,user,category_{KSAs}}, vector_{O,m}) \Rightarrow cor_{G_{PCH,user,category_{KSAs}},O,m}$$

Where,

$$\begin{aligned} G_{PCH,user,category_{KSAs}} &: G_{PCH,user} \text{ with values on vector } 0 \text{ except those of } category_{KSAs} \\ G_{PCH,user} &: \text{Gravity Center of Personal Career Histories } \{UDV_1^{user}\} \\ category_{KSAs} &\in \{ \text{Knowledge, Skills, Abilities} \} \end{aligned}$$

The function **III** requires input of a set of users' Personal Career Histories, and one of the $category_{KSAs}$. Personal Career Histories is expressed as a *User Description Vector* UDV_1 , Personal Knowledge, Skills, Abilities and Experiences. **III** prepares $G_{PCH,category_{KSAs}}$ and calculate cor .

When Knowledge is selected as $category_{KSAs}$, the function **III** outputs a set of occupations, which includes those related each other in terms of Knowledge. These occupations seems to be of the same or similar types of industries, where the same set of Knowledge is *Required*. However, not in the same way, when Skills is selected, the occupations included in the output does not seems to be of the same or similar types of industries, but seems to be a set of occupations that *Requires* free-of-industries, general Skills, that the user have already *Developed*. When Abilities is selected, the occupations included in the output seems to be a set of occupations that *Requires* free-of-industries, basic Abilities, that the user have already *Developed*.

4.4 **IV**. Discover career alternatives to your career goals in terms of KSAs

The function **IV** discovers career alternatives to your career goals in terms of KSAs, and the correlation computation function *correlation* is defined as follows.

$$correlation(vector_{O,k}, vector_{O,m}) \Rightarrow cor_{O,k,O,m}$$

By calculating correlation between Required KSAs for an Occupation (the Career Goal) and Required KSAs for all other Occupations, Occupational Alternatives for the Career Goal, as a set of occupations with ordered values of correlation, are acquired.

4.5 **V**. Design curriculums for acquiring important KSAs to realize career goals

The function **V** designs curriculums for acquiring important KSAs to realize career goals, and the correlation computation function *correlation* is defined as follows.

$$correlation(vector_{O,k}, vector_{EC,m}) \Rightarrow cor_{O,k,EC,m}$$

By calculating correlation between Required KSAs for an Occupation (the Career Goal) and Developable KSAs by an Educational Content, Occupational Alternatives for the Career Goal, as a set of educational contents with ordered values of correlation, are acquired.

4.6 VI. Design curriculums for acquiring missing KSAs to realize career goals

The function **VI** design curriculums for acquiring missing KSAs to realize career goals, and the correlation computation function *correlation* is defined as follows.

$$\text{correlation}(UDV_4, \text{vector}_{EC,m}) \Rightarrow \text{cor}_{UDV_4, EC,m}$$

UDV_4 is Required KSAs for Career Objectives, as shown in the Table 4. By calculating correlation between missing KSAs for an Occupation (the Career Goal) and Developable KSAs by an Educational Content, Curriculums for Acquiring Missing KSAs to Realize the Career Goal, as a set of educational contents with ordered values of correlation, are acquired.

5 Experiments

In this section, we perform a couple of experimental studies for evaluating the feasibility of a meta-level knowledge base system for discovering personal career opportunities, by demonstrating our method applying to the virtual users defined in 2.1 *Examples of Personal Career Development*.

5.1 Experimental Environments

We have introduced following existing databases for Occupations, Educational Contents, Social and Educational Issues and Personal Career Histories, and generated metadata, in the form of vectors, for the experiments.

LDB_O As a connected database of **Occupations**, we use results of the investigations by *O*NET*. *O*NET* offers another value, a statistical database of occupational titles. There exist 949 occupations (for version 12.0), with definitions of occupations and values of importance of all the KSAs. For generating vectors for Occupations, we retrieve each value of KSAs from the *Summary Report* of the *O*NET Online*.

LDB_{EC} As a connected database of **Educational Contents**, we have used lectures' database for Faculty of Policy Management and Faculty of Environment and Information Studies available at Shonan Fujisawa Campus, Keio University, *Keio SFC* [22], and selected 671 lectures, but the subjects of *Research Projects* (as their syllabuses in a different data format), *Language Communications* (as their syllabuses not written in English or in Japanese), and *Wellness* (as they are for practical skills of physical education).

LDB_{SEI} As a connected database for **Educational Issues**, we use documents which express *Clusters* for *Keio SFC* [22]. There exist 16 *Clusters*, and these are appropriate for expressing documents of Educational Issues. For example, we have a *Cluster* named "Bioinformatics" (abbreviated as BI), where students study genes, genomes, cells and life using computers.

LDB_{PCH} As a connected database for **Personal Career Histories**, we introduce virtual users describes in 2.1 *Examples of Personal Career Development*.

At **Phase-1**, Andrea has taken lectures of *Design Language*, *Media Technology Basic (Web)* and *Design Language Workshop (Formative Arts and Products)*, and At **Phase-2**, Barry has experienced occupations of *Computer Programmers* and *Database Administrators*.

KSAs included in each of $UDV_{1\text{Phase-1}}^{\text{Andrea}}$ for each of *category*^{KSAs} are as follows.

Knowledge = { Communications and Media, Computers and Electronics, Design, Education and Training, Engineering and Technology, Fine Arts, Mathematics }

Skills = { Mathematics, Writing }

Abilities = { Fluency of Ideas, Number Facility, Oral Expression, Originality, Visualization, Written Expression }

KSAs included in $UDV_{1Phase-2}^{Barry}$ for each of *category*^{*KSAs*} are as follows.

Knowledge = { Administration and Management, Clerical, Computers and Electronics, Customer and Personal Service, Economics and Accounting, Education and Training, English Language, Mathematics }

Skills = { Active Learning, Active Listening, Complex Problem Solving, Coordination, Critical Thinking, Instructing, Learning Strategies, Operations Analysis, Programming, Reading Comprehension, Technology Design, Time Management, Troubleshooting }

Abilities = { Deductive Reasoning, Flexibility of Closure, Inductive Reasoning, Information Ordering, Near Vision, Oral Comprehension, Oral Expression, Originality, Problem Sensitivity, Speech Clarity, Written Comprehension, Written Expression }

Queries are described as following two cases. The queries are corresponding to the virtual users' requirements for effective supports to discover personal career opportunities. The experimental results are expected to show superhuman solutions to the requirements.

Case-1: Andrea in College (**Phase-1** , **I**, **IV**, **V**, **VI**)

Case-2: Barry at Career Change (**Phase-2** , **I**, **II**, **III**)

5.2 Experimental Results

The results of **Case-1** are shown on the Tables 5,6,7,8, and the results of **Case-2** are shown on the Tables 9,10,11. Results that are higher than 10th in rank are shown.

Results of **Case-1** shows the process of personal career development by **Andrea** in college (**Phase-1**). **Andrea** is majoring **Design** in college (**Phase-1**) and has been active producing her works of **art**. She is thinking that she will not get a position in a company but wants to start working as a free-lance designer and to keep being an artist. Of course, she knows it is risky to be free-lance, and wants to generalize herself and her decision relatively in the whole job market.

Case-1 of **I** helps her discover career candidates that are appropriate to her learning histories (Table 5). *Designers*, *Artists* and related occupations such as *Architects* are shown just as her career histories tells. Because she has concrete career objectives, **IV** helps her generalize herself and her decision relatively in the whole job market, by discovering career alternatives to her career goals (Table 6). By **V**, she looks for lectures for acquiring important KSAs to realize her career goals (Table 7), and then she specifies lectures for acquiring missing KSAs to realize career goals by the results of **VI** (Table 8) Now, she selects her career with more conviction.

Results of **Case-2** shows the process of personal career development by **Barry** at his career change (**Phase-2**). **Barry** majored Database Technologies in college, has been working for a company as a **Database Administrator** and a **Computer Programmer**. Now, he wants different employment. He likes to explore possibilities of his career and find new opportunities by career analysis of what kind of occupation is appropriate for his career histories, what else are there any career opportunities if he learned additional Skills and Knowledge, and what kind of occupations are there if he does not limit his career only in the industry of Information and Communication Technologies.

Case-2 of **I** helps him discover career candidates that are appropriate to his working histories (Table 9). *Database Administrators*, *Computer Programmers*, and related career opportunities such as *Computer Science Teachers* are shown just as his career histories tells (Table 9). By the results of **II**, he discovers career opportunities as a list of occupational

candidates assuming that he learned *Biology* as an additional *ksa* (Table 10). Occupational candidates such as *Microbiologists* and *Epidemiologists* are shown in the list, which **Requires** knowledge of *Biology*, *Computers and Electronics* and *Mathematics*. By the results of **I I**, he discovers *crossover* career opportunities over the various types of industries. (Table 11). When Knowledge is selected as *category^{KSAs}*, occupational candidates which require Knowledge of *Computers and Electronics* and *Mathematics*, such as *Financial Analysts* and *Statistical Assistants*, are listed. When Skills is selected, occupations that require Skills of *Programing*, such as *Computer Software Engineers*, are listed. When Abilities is selected, occupations that require Oral and Written related Abilities, such as *Teachers* are listed. As for **Barry**'s case, a Skill of Programming is important requirement in the industrial field. On the other hand, selection of Knowledge as a *category^{KSAs}* helps **Barry** discover free-of-industries occupational candidates. Now, he discovers his career opportunities with more conviction.

5.3 Discussions

Personal career development changes drastically by introducing our system. So far the curriculum has been statically designed by the lecturers in the educational institutes. We are now able to design personalized curriculums for every individual student, in a dynamic way according to the situation of the students and the industry. Our learning opportunities are not limited only by the educational contents authorized by existing educational institutes, but also by every opportunities regardless of spatial and temporal limitations. For example, we can discover important lectures on web-based educational services, or in the video lecture archives in the libraries.

So far, our human abilities of collecting information and evaluating it appropriately for the objectives, have limited us from being free of tracing conventional ways of designing our career. We have designed our careers with limited experience and knowledge. Now we are able to explore our career opportunities out of comprehensive knowledge and information that are available via worldwide networks. We are able to analyze all the collected information using experts' knowledge related to career development.

This is to say that our system has ability of changing existing ways of career development to a knowledge-based innovative way.

6 Concluding Remarks

In this paper, we have presented an implementation method of a meta-level knowledge base system for analyzing personal career opportunities by connecting occupational and educational databases. This method is used to create dynamic relationship among heterogeneous databases on occupations, educational contents, social and educational issues and personal career information. In this method, several functions are defined for analyzing relationships among heterogeneous databases. These databases are connected and analyzed in a dynamic way according to users' contexts and situations. By using our method, individual career development becomes to be effectively supported when discovering personal career opportunities and designing personalized career development plans.

While we believes that our methods and applications are effective for people to discover personal career opportunities, there are many scopes of improvements. Each of databases might include more specified classification, such as *Mathmatics* of Knowledge includes *Statistics*. We will introduce other features than KSAs, if there exist suitable metrics for our system. We can add other functions on application by, such as, connecting to recruiting databases, and introducing legacy psychological tests for users' profiling. User Interfaces are also important for efficiently supporting users to discover personal career opportunities. The interfaces visualize results of computation on a spatial and temporal views, and show them adaptively as the users career are developed.

Table 5: The experimental results of **I** of Case-1: Our system discovers career candidates that are appropriate to **Andrea**'s career histories.

Rank	Correlation	Titles of Occupations
1	0.502459434369	Set and Exhibit Designers
2	0.496999544204	Engineering Teachers, Postsecondary
3	0.484182025907	Landscape Architects
4	0.467764276919	Architecture Teachers, Postsecondary
5	0.438529009633	Computer Software Engineers, Systems Software
6	0.43143348605	Film and Video Editors
7	0.424918292832	Multi-Media Artists and Animators
8	0.409293742348	Computer Hardware Engineers
9	0.40929374226	Atmospheric, Earth, Marine, and Space Sciences Teachers, Postsecondary
9	0.40929374226	Industrial Engineers

Table 6: The experimental results of **IV** of Case-1: Our system discovers career alternatives to **Andrea**'s career goals in terms of KSAs. *Art Directors* (left) and *Set and Exhibit Designers* (right) are selected as her career goals.

Rank	Correlation	Titles of Occupations	Rank	Correlation	Titles of Occupations
1	1	Art Directors	1	1	Set and Exhibit Designers
2	0.821952943376	Market Research Analysts	2	0.754851356096	Public Relations Specialists
3	0.786215858881	Public Relations Specialists	3	0.745241313525	Broadcast News Analysts
4	0.779773029164	Advertising and Promotions Managers	4	0.730769230769	Directors- Stage, Motion Pictures, Television, and Radio
5	0.758098043579	Reporters and Correspondents	5	0.704186850828	Landscape Architects
6	0.750478774386	Talent Directors	6	0.695181697093	Desktop Publishers
6	0.750478774386	Graphic Designers	7	0.692307692308	Editors
8	0.745869853983	Marketing Managers	8	0.691939188693	Art Directors
9	0.742781352708	Sales Representatives, Wholesale and Manufacturing, Technical and Scientific Products	9	0.680544653672	Actors
10	0.736955526661	Meeting and Convention Planners	9	0.680544653672	Reporters and Correspondents

Table 7: The experimental results of **V** of Case-1: Our system designs curriculums for acquiring important KSAs to realize **Andrea**'s career goals. *Art Directors* (left) and *Set and Exhibit Designers* (right) are selected as her career goals.

Rank	Correlation	Titles of Occupations	Rank	Correlation	Titles of Occupations
1	0.463523582362	Design Strategy (Ambient Media)	1	0.591312395989	Web Design and Its Management
2	0.454858826147	English Basics 2 (Gateway D)	2	0.566138517072	Writing Skills Workshop
2	0.454858826147	English Basics 2 (Gateway A)	3	0.558156305651	Writing Skills Workshop
2	0.454858826147	Design Strategy (Digital Sound)	4	0.522976360368	Programming Script Language (E)
2	0.454858826147	English Basics 2 (Gateway B)	5	0.518874521663	Design Language Workshop (Observation and Reception)
2	0.454858826147	English Basics 2 (Gateway C)	5	0.518874521663	Design Language Workshop (Observation and Reception)
7	0.415227399269	Presentation Skills	7	0.496138938357	Design Language Workshop (Information Design)
7	0.415227399269	Design Language Practice	7	0.496138938357	Design Language Workshop (Information Design)
7	0.415227399269	Design Strategy (Interaction)	9	0.489535463898	Design Strategy (Ambient Media)
7	0.415227399269	English Basics 1 (Gateway D)	10	0.485362671697	Entertainment Design

Table 8: The experimental results of **VI** of Case-1: Our system designs curriculums for acquiring missing KSAs to realize **Andrea**'s career goals. *Art Directors* (left) and *Set and Exhibit Designers* (right) are selected as her career goals.

Rank	Correlation	Titles of Occupations	Rank	Correlation	Titles of Occupations
1	0.310252613997	Marketing Strategy	1	0.35805743702	Design Strategy (Ambient Media)
2	0.307728727448	Legal Writing	2	0.346410161514	English Basic 1 (Gateway C)
3	0.301511344578	Consulting Skills	2	0.346410161514	English Basic 1 (Gateway D)
4	0.301511344578	PR Strategy	2	0.346410161514	English Basic 1 (Gateway B)
5	0.295656197995	Design Strategy (Ambient Media)	2	0.346410161514	English Basic 1 (Gateway A)
6	0.286038776774	English Basic 1 (Gateway D)	6	0.316227766017	English Basic 2 (Gateway A)
6	0.286038776774	English Basic 1 (Gateway C)	6	0.316227766017	English Basic 2 (Gateway C)
6	0.286038776774	English Basic 1 (Gateway A)	6	0.316227766017	English Basic 2 (Gateway B)
6	0.286038776774	English Basic 1 (Gateway B)	6	0.316227766017	English Basic 2 (Gateway D)
10	0.269679944985	Administration Analysis	10	0.298142397	Risk and Security

Table 9: The experimental results of **I** of Case-2: Our system discovers career candidates that are appropriate to **Barry**'s career histories.

Rank	Correlation	Titles of Occupations
1	0.932007033244	Database Administrators
2	0.91655875972	Computer Programmers
3	0.79336327578	Tax Examiners, Collectors, and Revenue Agents
4	0.790179876011	Computer Science Teachers, Postsecondary
4	0.790179876011	Network and Computer Systems Administrators
4	0.790179876011	Network Systems and Data Communications Analysts
7	0.784046983414	Computer Systems Analysts
7	0.784046983414	Financial Examiners
9	0.782960292686	Computer Software Engineers, Systems Software
9	0.782960292686	Computer Hardware Engineers

Table 10: The experimental results of **II** of Case-2: Our system discovers career opportunities (occupational candidates) assuming that **Barry** acquires an additional *ksa*, *Biology*. Occupations such as *Microbiologists* and *Epidemiologists* are shown up which are not shown in the results of **I** of Case-2 (Table 9).

Rank	Correlation	Titles of Occupations
1	0.911293179513	Database Administrators
2	0.896188243825	Computer Programmers
3	0.796696475909	Microbiologists
4	0.775730779174	Tax Examiners, Collectors, and Revenue Agents
5	0.772618130457	Network Systems and Data Communications Analysts
5	0.772618130457	Network and Computer Systems Administrators
5	0.772618130457	Computer Science Teachers, Postsecondary
8	0.76662154138	Computer Systems Analysts
8	0.76662154138	Financial Examiners
10	0.765559002351	Epidemiologists

Table 11: The experimental results of **III** of Case-2: Our system discovers *crossover* career opportunities for **Barry** over the various types of industries.

Rank	Titles of Occupations (category _{KSA} s = Knowledge)	Rank	Titles of Occupations (category _{KSA} s = Skills)	Rank	Titles of Occupations (category _{KSA} s = Abilities)
1	Database Administrators	1	Computer Programmers	1	Environmental Compliance Inspectors
2	Financial Examiners	1	Database Administrators	1	Computer Science Teachers, Postsecondary
3	Insurance Adjusters, Examiners, and Investigators	3	Computer Hardware Engineers	1	Market Research Analysts
3	Tax Preparers	4	Architectural Drafters	1	Medical Scientists, Except Epidemiologists
3	Financial Analysts	4	Cartographers and Photogrammetrists	1	Electrical Drafters
3	Auditors	4	Computer Software Engineers, Applications	1	Health Specialties Teachers, Postsecondary
3	Statistical Assistants	4	Engineering Managers	1	Environmental Engineering Technicians
3	Procurement Clerks	4	Computer Software Engineers, Systems Software	1	Atmospheric and Space Scientists
3	Bookkeeping, Accounting, and Auditing Clerks	4	Aerospace Engineering and Operations Technicians	1	Power Distributors and Dispatchers
10	First-Line Supervisors/Managers of Office and Administrative Support Workers	4	Computer Support Specialists	1	Airfield Operations Specialists

Acknowledgement

This research was partially supported by the Ministry of Education, Science, Sports and Culture of Japan, Grant-in-Aid for Young Scientists (B), 18700097, from 2006 to 2008. We would like to express our thanks to Dr. Ikuyo Kaneko, Dean and Professor of Graduate School of Media and Governance, Keio University, Japan, and Dr. Koichi Furukawa, Professor of Graduate School of Media and Governance and Faculty of Environment and Information Studies, Keio University, Japan, for valuable comments. We are grateful for their support.

References

- [1] Takahashi, Y. and Kiyoki, Y., “A Meta-Level Integration Method for Job Information Databases (in Japanese),” DBSJ Letters Vol.2 No.3, pp.33–36, 2003.
- [2] Takahashi, Y. and Kiyoki, Y., “The Implementation and Application of a Meta-Level Career-Development Support System,” In Proceedings of The 7th IASTED International Conference on COMPUTERS AND ADVANCED TECHNOLOGY IN EDUCATION (CATE 2004), Hawaii, USA, pp.558–563, 2004.
- [3] Takahashi, Y. and Kiyoki, Y., “A Meta-Level Database System Connecting Occupational and Educational Databases (in Japanese),” DBSJ Letters, Vol.5, No.4, pp.29–32, March 2007.
- [4] RECRUIT CO., LTD., “rikunabi,” URL: <http://www.rikunabi.com/>.
- [5] The Japan Institute for Labour Policy, “CAREER MATRIX,” URL: <http://cmx.vrsys.net/>.
- [6] Kanai, T., “Career Design for Working People,” PHP Kenkyujo, 2002 (in Japanese , Japanese Title is “Hataraku Hito No Career Design”).
- [7] Okubo, T., “A Guide to Career Design, 1, Basic Skills,” Nikkei Inc., 2006 (in Japanese , Japanese Title is “Career Design Nyumon, 1, Kisoryoku Hen”).
- [8] Bright, M. W., Hurson, A. R. and Pakzad, S. H., “A Taxonomy and Current Issues in Multidatabase System,” IEEE Computer, Vol.25, No.3, pp.50–59, 1992.
- [9] Litwin, W., Mark, L. and Roussopoulos, N., “Interoperability of Multiple Autonomous Databases,” ACM Computing Surveys, Vol.22, No.3, pp.267–293, 1990.
- [10] Sheth, A. and Larson, J.A., “Federated database systems for managing distributed, heterogeneous, and autonomous databases, ACM Computing Surveys, Vol.22, No.3, pp.183–236, 1990.
- [11] Kiyoki, Y. and Kitagawa, T. “A metadatabase system supporting interoperability in multidatabases,” Information Modeling and Knowledge Bases, Vol.5, pp.287–298, 1993.
- [12] Kiyoki, Y., Kitagawa, T. and Hitomi, Y., “A fundamental framework for realizing semantic interoperability in a multidatabase environment,” Journal of Integrated Computer-Aided Engineering, Vol.2, No.1, pp.3–20, 1995.
- [13] Kiyoki, Y., Hosokawa, Y. and Ishibashi, N., “A Metadatabase System Architecture for Integrating Heterogeneous Databases with Temporal and Spatial Operations,” Advanced Database Research and Development Series Vol. 10, Advances in Multimedia and Databases for the New Century, A Swiss/Japanese Perspective, World Scientific Publishing, pp.158–165, 1999.

- [14] Hosokawa, Y., Ishibashi, N., Yashiro, Y. and Kiyoki, Y., “A Data Integration Method Realizing Evaluation for Temporal and Spatial Relationships in a Multidatabase Environment,” IPSJ Transactions on Databases, Vol.40, No.SIG 8(TOD4), pp.95–111, 1999.
- [15] Salton, G., Wong, A. and Yang, C. S., “A vector space model for automatic indexing,” Communications of the ACM, Vol. 18, No. 11, pp.613–620, 1975.
- [16] Michael W. Berry and Susan T. Dumais, and Gavin W. O’Brien, “Using Linear Algebra for Intelligent Information Retrieval,” December 1994. Published in SIAM Review 37:4, pp. 573–595, 1995.
- [17] M.W. Berry, Z. Drmac, and E.R. Jessup, “Matrices, Vector Spaces, and Information Retrieval,” SIAM Review 41:2, pp.335–362, 1999.
- [18] Project Management Institute, “A Guide to the Project Management Body of Knowledge (PMBOK Guide)–2000 Edition,” 2001.
- [19] ACM/IEEE Computer Society Committee, “Guide to the Software Engineering Body of Knowledge,” URL: <http://www.swebok.org/>.
- [20] The Joint Task Force on Computing Curricula IEEE Computer Society and Association for Computing Machinery, “Computing curricula 2001,” Journal on Educational Resources in Computing (JERIC), Volume 1, Issue 3es, No.1, 2001.
- [21] The Occupational Information Network (O*NET), URL: <http://www.onetcenter.org/>.
- [22] Shonan Fujisawa Campus, Keio University (Keio SFC), URL: <http://www.sfc.keio.ac.jp/>.

Temporal Entities in the Context of Cross-Cultural Meetings and Negotiations

Anneli HEIMBÜRGER

University of Jyväskylä

Faculty of Information Technology

Information Technology Research Institute

P.O. Box 35 (Agora)

FIN-40014 University of Jyväskylä, Finland

anneli.heimbürger@titu.jyu.fi

Abstract. Time is an essential dimension in our knowledge space. Understanding different temporal dimensions and dynamics in cross-cultural meetings and negotiation processes will improve our skills in cross-cultural communication and increase our cultural competence. It also helps us to identify and formalize cross-cultural concepts and related temporal entities and to construct for example cross-cultural XML Schemas. We address three issues in our paper. First, we discuss implications of cross-cultural differences for meetings and negotiations and introduce the concept of a temporal entity in this context. Second, we present three relevant ontological approaches – OWL-Time, temporal aggregates and temporal regions of the Span ontology – for modelling temporal entities in the context of cross-cultural meetings and negotiations. Third, we give proof-of-concept examples of applying those ontological approaches to scheduling meetings and recurrent actions over time zones and to describing temporal parts of cross-cultural meetings and negotiations between Finland and Japan. Cross-cultural XML Schemas and temporal entities have important roles in design and implementation of culture-sensitive information systems.

Keywords. Culture-Sensitive Information Systems, CSIS, Cross-Cultural Meetings and Negotiations, Temporal Entities, OWL-Time, Temporal Aggregates, Perdurants, BFO/Span Ontology, Temporal Regions of Time

1. Introduction

Globalization is one of the main trends in our world. Increasingly, eastern and western cultures meet each other in connection with business, research, governmental activities, environmental protection, emergency situations, higher education, medical and social care and tourism. Cross-cultural actions are carried out both in virtual world and in physical world like via e-mails, Web meetings, and collaborative virtual working spaces as well as in face-to-face meetings. In situations like these, we may experience some cultural differences relating to business and social etiquette, structure of language and structure of thinking, attitudes towards time, personal space, face-saving, communication style, acceptance and use of silence.

Cultural sensitivity has become an important dimension for success in today's international business and research arena. Despite the trend of globalization, business

executives, project managers and project team members are finding themselves in uncertain situations due to culturally dependent differences in communication protocol, language and value systems. Consequently, people involved in cross-cultural transactions are advised to be aware of the cultural backgrounds of their counterparts. Cultural competence might help for example project managers to achieve project goals and avoid potential risks in cross-cultural project environments and would also support them to promote creativity and motivation through flexible leadership. By understanding the main cultural dimensions and by adjusting to cultural differences, people can become better negotiators, project managers and research team members on behalf of their companies and research organizations.

The main objective of our research project is to design and implement a culture-sensitive information system (CSIS) that provides cultural assistant for people attending in cross-cultural meetings, participating in cross-cultural negotiation processes or for people working in cross-cultural projects [8]. For realizing CSIS we have to test different kinds of approaches, models, methods and technologies. The whole system includes several components (Figure 1). The focus of our paper is on collaborative cross-cultural meetings and negotiations and temporal concepts related to them. On the one hand, from the viewpoint of cross-cultural meetings and negotiations, they involve teams and individuals with different cultural backgrounds and also with different sense of time and attitudes towards meetings and negotiation processes. On the other hand, from the viewpoint of time, there are several temporal processes related to cross-cultural meetings and negotiations.

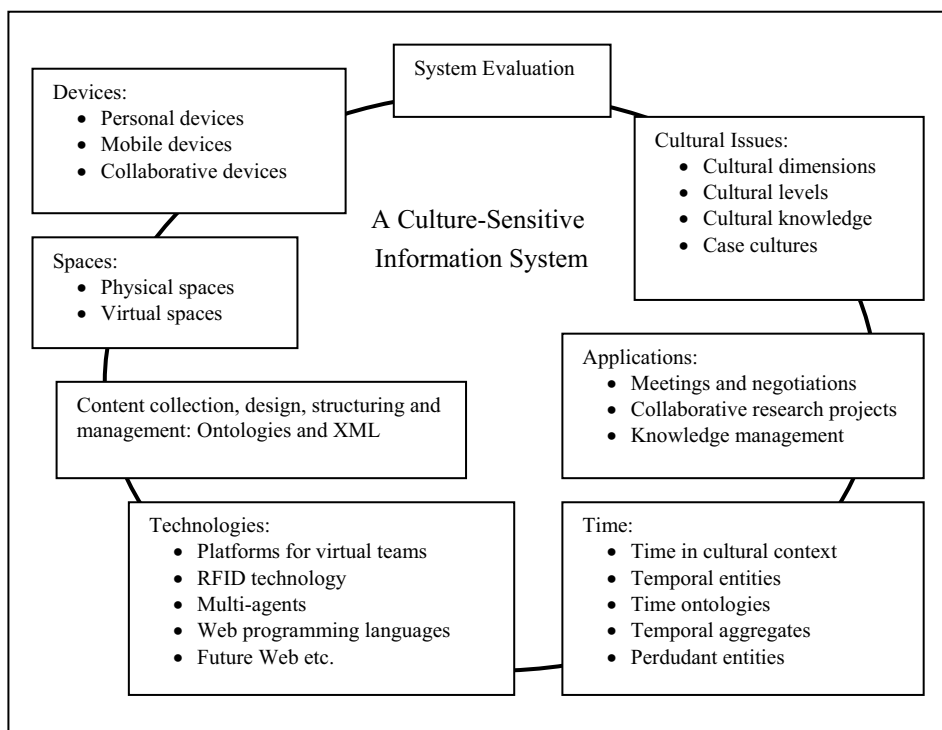


Figure 1. Basic components of a culture-sensitive information system

According to King [14] cultures can be considered at five levels: (1) national cultures, (2) organizational cultures, (3) organizational subcultures, (4) subunit cultures and (5) team cultures. Our paper concentrates on the national culture level. Case cultures in our study are Finnish and Japanese. The contribution of the paper is to (a) identify different kinds of temporal entities in the context of cross-cultural meetings and negotiations, (b) describe relevant ontological approaches related to these temporal entities and (c) illustrate some proof-of-concept applications. The study is exploratory and a small vertical part of the whole horizontal CSIS designing process. The study also aims to discuss on the concept of the perdurant entity in the cross-cultural context. We have analyzed meeting and negotiation processes and identified fourteen conceptual main classes that are dependent on time. Our analysis is based on interviewing Finnish companies working in Japan.

The remainder of the paper is organized as follows. In Section 2, we describe some implications of cross-cultural differences for meetings and negotiations. In Section 3, we introduce the concept temporal entity. In Section 4, we describe OWL-Time and its application to scheduling meetings across time zones. In Section 5, we describe an extension of OWL-Time to represent temporal aggregates and its applications. In Section 6, we discuss the concept of a perdurant entity and Span ontology, the temporal component of the Basic Formal Ontology. In Section 7, we apply Span approach to represent temporal entities of cross-cultural meetings and negotiations. Section 8 is reserved for conclusions and issues for further research.

2. Implications of Cross-Cultural Differences for Meetings and Negotiations

The last decade has seen a dramatic increase in the importance of cross-cultural meetings and negotiations. Examples of activities are business operations and decision making in boards of cross-cultural companies, negotiations concerning bilateral research agreements, preparation and implementation of collaborative research projects. Some examples of general subject classes related to meetings and negotiations where differences between eastern and western cultures usually occur are given in Table 1.

Table 1. Examples of general subject classes where eastern and western negotiators may face differences

<ul style="list-style-type: none">• The Language of Collaboration• Build a Relationship Before• Meeting Protocol• Hierarchy, Status and Gender• Concern with Face• Maintaining Surface Harmony• Formality and Rituals• Orientation to Time	<ul style="list-style-type: none">• Communication Style• Making a Presentation• Determining the Bargaining• Decision-Making Behavior• Role of the Contract• Gift Giving and Receiving• Wining and Dining• Maintaining the Relationships
---	--

There exist several studies and conceptual categories for assessing cultures [3, 7, 10, 11, 12, 13, 15]. These studies consider role differentiation according to age, gender and hierarchy, orientation towards risks, shared knowledge, beliefs and rules of logical thinking, shared goals, attitudes to time, and attitudes to environments. Hofstede’s

framework for assessing cultures is one of the widely used and cited frameworks in the context of cross-cultural research [10, 11].

Hofstede's approach proposes a set of cultural dimensions along which dominant value systems can be ordered. The framework consists of five dimensions: individualism/collectivism, power distance, masculinity/femininity, uncertainty avoidance and long-term orientation/short-term orientation. All dimensions are generalizations and individuals may vary from their society's descriptors. The scores of cultural dimensions in different countries according to Hofstede's research are given in [12]. The survey is extensively described in [10]. The figures should not be taken literally. However they provide interesting information because they show differences in answers between groups of respondents. Different value systems affect human thinking, feeling, and acting, and the behavior of teams and organizations as well as the temporal dimensions of research projects and negotiations. In Table 2, we have summarized implications of cross-cultural differences for meetings and negotiations according to Hofstede's dimensions of culture [18].

Table 2. Implications of cross-cultural differences for meetings and negotiations

Dimension	Implication
Individualism/ Collectivism	Negotiators from a collectivistic society are likely to spend more time on long-term goals, are more likely to make realistic offers, and are more likely to be cooperative. Conversely, negotiators from individualistic societies are more likely to focus on the short-term, make extreme offers, are more likely to view negotiations from a fixed perspective, and are more likely to be competitive.
Power distance	Negotiators from low power distance cultures may be frustrated by the need of negotiators from high power distance cultures to seek approvals from higher authority. On the other hand, negotiators from high power distance cultures may feel pressured by the pace imposed by negotiators from low power distance cultures.
Masculinity/ Femininity	When negotiating, individuals from masculine cultures are more likely to be competitive (win-lose) and those from feminine cultures to be empathic and seek compromise (win-win). This means that negotiators from masculine cultures are likely to view the feminine negotiator as "avoiding" while the feminine negotiator is likely to view their masculine negotiator as "contending."
Uncertainty avoidance	Negotiators from high risk avoidance cultures are likely to view those from low risk avoidance cultures as unfocused. Those from low risk avoidance cultures are likely to view negotiators from high risk avoidance cultures as rigid.
Long-term/short- term orientation	Long-term/short-term orientation refers to the extent to which a culture programs its members to accept delayed gratification of their material, social, and emotional needs. Business people in long-term oriented cultures are accustomed to working toward building strong positions in their markets and do not expect immediate results. In short-term oriented cultures the results of the past month, quarter, or year is a major concern. Time is seen in a different way by eastern and western cultures and even within these groupings temporal culture differs from country to country. Also temporal identities of different organizations and teams in organizations may vary.

All dimensions include more or less embedded references to long-term orientation and to short-term orientation. In the following sections we will analyze the long-term – short term continuum in more detail and try to identify its subsets which, we, in this study, call temporal entities.

3. Temporal Entities

A temporal entity describes a point in time, event, or time period at a conceptual level. Objects, such as people, are not temporal entities. However, they can have a time-of-existence property which is a temporal entity, in particular, related to situations.

The identification of temporal entities depends on the context. The identification approach is based on named entity extraction that determines temporal expressions. A temporal expression is basically a sequence of tokens that represent an instance of a temporal entity. If the context is knowledge management the identification involves a linguistic analysis of the knowledge. If the context is a situation or a process such as a meeting or a negotiation process, the identification of temporal entities involves a progressive analysis of the situation or the process. Similar to the approach by Alonso, Gertz and Baeza-Yates [2], we identify the following three main categories of temporal expressions in our study.

- Category 1. Explicit temporal expressions: These temporal expressions directly describe entries in some timeline, such as an exact date or year. For example, the token sequences “January 2009” or “September 14, 2008, 3.00 p.m.” are explicit temporal expressions and can be mapped directly to a timeline.
- Category 2. Implicit temporal expressions: Depending on the underlying time ontology and capabilities of the named entity extraction approach, even apparently imprecise temporal information, such as names of holidays or events can be anchored in a timeline. For example, the token sequence “Ocean Day 2008 in Japan” can be mapped to the expression “July 21, 2008”, or the sequence “Midsummer Day 2008 in Finland” can be mapped to “June 21, 2008”. Implicit temporal expressions can also be collections of temporal entities such as “every other Wednesday in every second month”.
- Category 3. Relative temporal expressions: These temporal expressions represent temporal entities that can only be anchored in a timeline in reference to another explicit or implicit, already anchored temporal expression (for example a starting time of a meeting). For example, the expression “3 p.m.” alone cannot be anchored in any timeline. However, it can be anchored if the date of the meeting is known. This date then can be used as a reference for that expression, which then can be mapped to a timeline. Allen’s temporal calculus can be used to express relationships between relative temporal expressions [1].

In the following three sections we study ontological approaches related to these categories and their proof-of-concept applications respectively in the context of cross-cultural meetings and negotiations. In Table 3, a practical example, a theoretical

approach and an example of a proof-of-concept implementation is given related to three categories of temporal entities.

Table 3. Practical examples, theoretical approaches and examples of a proof-of-concept implementation related to categories of temporal entities

Category	Practical Example	Theoretical Approach	Example of a Proof-of-Concept Implementation
Cat1	Scheduling a cross-cultural Web meeting over time zones	OWL-Time	Temporal reasoner component of the CSIS system for determining appropriate meeting times among the project team members.
Cat1 + Cat2	Scheduling collaborative recurrent actions between project team members	Temporal aggregates: An extension of the OWL-Time	Extended temporal reasoner component of the CSIS system for determining appropriate recurrent meeting times among the project team members.
Cat3	Identifying temporal entities related to cross-cultural meetings and negotiations	The concept of perdurant entities in general and Span ontology, which is the temporal component of the Basic Formal Ontology	One of the educational components of the CSIS system for project managers and team members could illustrate the temporal progress of cross-cultural meetings and negotiations.

4. OWL-Time: An Ontology of Temporal Concepts

OWL-Time is an ontology of temporal concepts [21]. The ontology provides a vocabulary for expressing facts about topological relations among instants and intervals, together with information about durations, date times and time zones. From our research point of view, the most essential concepts of the OWL-Time are shortly summarized here [21].

4.1 Topological Temporal Relations

OWL-Time defines two subclasses of TemporalEntity: Instant and Interval. Instants are things with extent and instants are point-like things. An instant is an interval with zero length.

```
( $\forall$  T) [TemporalEntity(T)  $\equiv$  Interval(T)  $\vee$  Instant(T)]
:TemporalEntity
  a owl:Class ;
  rdfs:subClassOf :TemporalThing ;
  owl:equivalentClass
    [ a owl:Class ;
      owl:unionOf (:Instant :Interval)
    ] .
```

The predicates “begins” and “ends” are relations between instants and temporal entities.

```

begins(t, T)  $\supset$  Instant(t)  $\wedge$  TemporalEntity(T)
:hasBeginning
  a      owl:ObjectProperty ;
  rdfs:domain :TemporalEntity ;
  rdfs:range :Instant .

```

```

ends(t, T)  $\supset$  Instant(t)  $\wedge$  TemporalEntity(T)

```

A proper interval can be defined as an interval whose start and end are not identical.

```

( $\forall$  T) [ProperInterval(T)  $\equiv$  Interval(T)
 $\wedge$  ( $\forall$  t1, t2) [begins(t1, T)  $\wedge$  ends(t2, T)  $\supset$  t1  $\neq$  t2]]
:ProperInterval
  a      owl:Class ;
  rdfs:subClassOf :Interval ;
  owl:disjointWith :Instant .

```

The “before” relation of temporal entities gives directionality to time. If a temporal entity T₁ is before another temporal entity T₂, then the end of T₁ is before the beginning of T₂. Thus, “before” can be considered to be basic to instants and derived for intervals.

```

( $\forall$  T1, T2) [before(T1, T2)  $\equiv$  ( $\exists$  t1, t2) [ends(t1, T1)  $\wedge$  begins(t2, T2)  $\wedge$  before(t1, t2)]]

```

The interval relations of Allen’s temporal interval calculus [1] can be defined in terms of “before” and identity on the beginning and end points. OWL-Time provides the interval relations: intervalEquals, intervalBefore, intervalMeets, intervalOverlaps, intervalStarts, intervalDuring, intervalFinishes, and their reverse interval relations. For example, the specification of intervalEquals is:

```

( $\forall$  T1, T2) [intEquals(T1, T2)  $\equiv$  [ProperInterval(T1)  $\wedge$ 
ProperInterval(T2)  $\wedge$  ( $\forall$  t1) [begins(t1, T1)  $\equiv$  begins(t1, T2)]  $\wedge$ 
( $\forall$  t2) [ends(t2, T1)  $\equiv$  ends(t2, T2)]]]
:intervalEquals
  a      owl:ObjectProperty ;
  rdfs:domain :ProperInterval ;
  rdfs:range :ProperInterval .

```

An interval can have one duration but several duration descriptions, such as 1 day or 24 hours. OWL-Time defines a specific kind of individual called a “Duration Description” with property values years, months, weeks, days, hours, minutes, and seconds (the following syntax is only an example part of DurationDescription):

```

:DurationDescription
  a      owl:Class ;
  ...
  rdfs:subClassOf
    [ a      owl:Restriction ;
      owl:maxCardinality 1 ;
      owl:onProperty :hours
    ] ;

```

For a date time description in OWL-Time, xsdDateTime relation can be defined. The relation uses the XML Schema datatype dateTime as its range. For example, an instant represents the start of a meeting, called meetingStart, and it happens at 10:30am Japan time on 24/01/2009 can be expressed using inXSDDateTime in OWL as:

```

:meetingStart
  a      :Instant ;
  :inDateTime
        :meetingStartDescription ;
  :inXSDDateTime
        2009-01-24T10:30:00+9:00 .

```

4.2 An Example of a Proof-of-Concept Application: Scheduling a Cross-Cultural Web Meeting over Time Zones

As an example of applying OWL-Time we study scheduling a cross-cultural research project meeting over time zones. Let's suppose that the project manager has a telecon scheduled for 6:00 pm (Japan time) on September 14, 2008. The other researcher would like to have a Web meeting with him/her for 2:00 pm (Finnish time) on the same day, and expect the Web meeting to last 45 minutes. Will there be an overlap with the telecon and the proposed Web meeting? In this case, the facts about the telecon and the Web meeting can be specified by using OWL-Time that will allow a temporal reasoner to determine whether there is a conflict (Figure 2).

```

:telecon
  a      :Interval ;
  :hasBeginning :teleconStart .
:meeting
  a      :Interval ;
  :hasBeginning :meetingStart ;
  :hasDurationDescription
        :meetingDuration .
:teleconStart
  a      :Instant ;
  :inXSDDateTime
        2008-09-14T18:00:00+9:00 .
:meetingStart
  a      :Instant ;
  :inXSDDateTime
        2008-09-14T14:00:00+3:00 .
:meetingDuration
  a      :DurationDescription ;
  :minutes 45 .

```

Figure 2. An OWL-Time example of scheduling a cross-cultural meeting over time zones

The telecon and the meeting are defined as intervals. The start times of the telecon and the meeting is specified by using `teleconStart` and `meetingStart` respectively. The date times are specified using `inXSDDateTime`. The duration of the meeting is specified using the duration description class. In practice, a temporal reasoner could be a component of a culture-sensitive information system (CSIS) for determining appropriate meeting times among the project team members.

5. Temporal Aggregates: An Extension of the OWL-Time

The OWL-Time includes the topological aspects of time, measures of duration, and the date times and time zone information. Temporal aggregates are collections of temporal entities. Examples of temporal aggregates are “every 3rd Wednesday in 2008”, and “3 consecutive Tuesdays”. Temporal aggregates are very common in collaborative research project functions such as in management, in organizing recurrent meetings, in progress monitoring and in scheduling team work and related deliverables. OWL-Time has been extended to cover temporal aggregates as well [9]. We summarize some essential concepts here [19].

5.1 Temporal Sequences, Their Members and Temporal Aggregate Descriptions

The same temporal aggregate can be broken up into a set of intervals in many different ways. A minimal temporal sequence is one whose intervals are maximal, so that the number of intervals is minimal. For example, a week can be viewed as a week or as 7 individual successive days. The first option would be minimal. We can go from a non-minimal to a minimal temporal sequence by concatenating intervals that meet.

In order to encode the temporal aggregates ontology in OWL, temporal sequence, temporal sequence member and temporal aggregate description are defined. Temporal sequence has only one optional property `hasMember` which maps from a temporal sequence to any temporal thing:

```
<owl:Class rdf:ID="TemporalSeq">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMember"
      />
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">0
      />
    />
  />
</owl:Class>

<owl:ObjectProperty rdf:ID="hasMember">
  <rdfs:domain rdf:resource="#TemporalSeq" />
  <rdfs:range rdf:resource="#TemporalThing" />
</owl:ObjectProperty>
```

`TemporalSeqMember` points from the temporal sequence member to its associated sequence. It has a required pair of properties: `isMemberOf` and `hasPosition`, so that it can point back to the associated sequence and also locate itself in the sequence:

```
<owl:Class rdf:ID="TemporalSeqMember">
  <rdfs:subClassOf rdf:resource="#TemporalThing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#isMemberOf"
      />
      <owl:cardinality
        rdf:datatype="&xsd;nonNegativeInteger">1
      />
    />
  />
```

```

        </owl:cardinality>
    </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="#hasPosition"
        />
        <owl:cardinality
            rdf:datatype="&xsd;nonNegativeInteger"
            r">1
        </owl:cardinality>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="isMemberOf">
    <rdfs:domain rdf:resource="#TemporalSeqMember" />
    <rdfs:range rdf:resource="#TemporalSeq" />
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="hasPosition">
    <rdfs:range rdf:resource="&xsd;integer" />
</owl:DatatypeProperty>

```

The most essential class of the temporal aggregates ontology in OWL is the temporal aggregate description class. It specifies the temporal aggregate description for temporal sequences. It is associated with the temporal sequence class by `hasTemporalAggregateDescription`-property.

```

<owl:Class rdf:ID="TemporalAggregateDescription">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasStart" />
            <owl:maxCardinality
                rdf:datatype="&xsd;nonNegativeInteger">1
            </owl:maxCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    ...
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasCount" />
            <owl:maxCardinality
                rdf:datatype="&xsd;nonNegativeInteger">1
            </owl:maxCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="hasTemporalAggregateDescription">
    <rdfs:domain rdf:resource="#TemporalSeq" />
    <rdfs:range rdf:resource="#TemporalAggregateDescription"
    />
</owl:ObjectProperty>

```

The properties of the temporal aggregate description with examples are summarized in Table 4.

Table 4. The properties of a temporal aggregate description class

Property	Mapping from the temporal aggregate description (an example)
hasStart, hasEnd	Maps to the instant thing specifying the start and the end instants of a temporal sequence (times and dates in a calendar and clock)
hasContextTemporalSeq	Maps to the temporal sequence specifying the context (super) temporal sequence of a given (sub) temporal sequence (every Wednesday)
hasithTemporalUnit	Maps to positive integers specifying the i^{th} temporal unit elements in the temporal sequence (every 5 th Wednesday, Thursday and Saturday)
hasTemporalUnit hasContextTemporalUnit	Maps to the temporal unit (January 24 th 2009)
hasPosition	Maps to integers specifying the position of the element in the temporal sequence (the first two Tuesdays in every June, hasPosition = 2)
hasGap	Maps to positive integers specifying the gap between the elements in the temporal sequence (every 5 th Tuesday, hasGap = 5)
hasCount	Maps to positive integers specifying the cardinality or the size of the temporal sequence (three consecutive Mondays, hasCount = 3)

5.2 *An Example of a Proof-of-Concept Application: Scheduling Recurrent Actions*

Temporal aggregate ontology can be used to represent complex multiple-layered collections of temporal entities for scheduling collaborative research activities such as "every other week on Tuesday, Wednesday and Friday until September 14, 2008, starting on Thursday, January 24, 2008" or "every other Monday in every 5th month". An example in OWL of the expression "every other week on Monday until January 24, 2009, starting on Sunday September 14, 2008" is given in Figure 3.

```
<time-entry:TemporalSeq rdf:ID="tseq">
  <time-entry:hasTemporalAggregateDescription
    rdf:resource="#MeveryOtherWeek" />
</time-entry:TemporalSeq>

<time-entry:TemporalSeq rdf:ID="tseq-everyOtherWeek">
  <time-entry:hasTemporalAggregateDescription
    rdf:resource="#everyOtherWeek" />
</time-entry:TemporalSeq>

<time-entry:TemporalAggregateDescription
  rdf:ID="everyOtherWeek">
  <time-entry:hasTemporalUnit rdf:resource="&time-
    entry;unitWeek" />
  <time-entry:hasGap rdf:datatype="&xsd;positiveInteger">2
  </time-entry:hasGap>
</time-entry:TemporalAggregateDescription>

<time-entry:TemporalAggregateDescription
```

```

        rdf:ID="MeveryOtherWeek">
    <time-entry:hasStart rdf:resource="#tseqStart" />
    <time-entry:hasEnd rdf:resource="#tseqUntil" />
    <time-entry:hasContextTemporalSeq rdf:resource="#tseq-
everyOtherWeek" />
    <time-entry:hasithTemporalUnit
    rdf:datatype="&xsd;positiveInteger">1
    </time-entry:hasithTemporalUnit>
    <time-entry:hasTemporalUnit rdf:resource="&time-
entry;unitDay" />
    <time-entry:hasContextTemporalUnit rdf:resource="&time-
entry;unitWeek" />
</time-entry:TemporalAggregateDescription>

<time-entry:Instant rdf:ID="tseqStart">
    <time-entry:inCalendarClock
    rdf:resource="#tseqStartDescription" />
</time-entry:Instant>

<time-entry:Instant rdf:ID="tseqUntil">
    <time-entry:inCalendarClockDataType
    rdf:datatype="&xsd;dateTime">2009-01-24
    </time-entry:inCalendarClockDataType>
</time-entry:Instant>

<time-entry:CalendarClockDescription
    rdf:ID="tseqStartDescription">
    <time-entry:unitType rdf:resource="&time-entry;unitDay" />
    <time-entry:year rdf:datatype="&xsd;gYear">2008
    </time-entry:year>
    <time-entry:month rdf:datatype="&xsd;gMonth">9
    </time-entry:month>
    <time-entry:day rdf:datatype="&xsd;gDay">14</time-
entry:day>
    <time-entry:dayOfWeekField
    rdf:datatype="&xsd;nonNegativeInteger">7
    </time-entry:dayOfWeekField>
</time-entry:CalendarClockDescription>

```

Figure 3. An example of a multiple-layered temporal aggregate

Temporal aggregate ontology can also be used to represent conditional temporal aggregates such as "Every Friday that's a holiday in Japan or in Finland". In practice, an extended temporal reasoner could be a component of the CSIS system for determining appropriate recurrent meeting times among the project team.

6. Endurants and Perdurants

The difference in terminology used between separate formal upper level ontologies can be quite substantial. However the one and foremost dichotomy most formal upper level ontologies apply is that between "endurants" and "perdurants" [17]. Endurants are wholly present i.e., all their proper parts are present at any time they are present. Endurants are entities that can be observed as a complete concept, at no matter which given snapshot of time. Examples are material objects, such as an apple or human and abstract objects, such as an organization or the border of a country.

Perdurants, on the other hand, are entities that are only partially present, in the sense that some of their proper temporal parts (e.g., their previous or future phases) may be not present. Perdurants are those entities for which only a part exists if we look at them at any given snapshot in time. When we freeze time we can only see a part of

the perdurant. Perdurants are often what we know as processes, for example “negotiating”. If we freeze time then we only observe a part of the negotiating. Without any previous knowledge we might not even be able to determine the actual process as being a process of negotiating.

Basic Formal Ontology (BFO) is a theory of the basic structures of reality [17, 20]. It is developed at the Institute for Formal Ontology and Medical Information Science (IFOMIS) in the University of Leipzig, Germany. The Basic Formal Ontology (BFO) has two components: a Snap ontology and a Span ontology. The Snap ontology of endurants is used to characterize static views of the world. Snap requires a temporal logic of a certain grade if it would be used in temporal contexts.

The other component of BFO is the Span ontology. Span is an ontology of occurrences and, more generally, an ontology of entities which have temporal parts. Span entities are divided mainly into (a) processual entities which are happenings or occurring entities, changes of various kinds in substantial entities, (b) spatio-temporal regions i.e. four dimensional regions of space-time and (c) temporal regions i.e. the whole of time and all of its parts [18, 20]. From the viewpoint of our study we focus on temporal regions of Span.

Time is the maximal temporal region, and it is a perdurant, and thus a Span entity. A temporal region is a part of time. Instants of time are zero-dimensional boundaries of extended temporal regions. In Span, temporal order reflects the structure of time as a linear continuum, and uses as a primitive the relation “before”. Relation “before” is a strict total order which holds between two time instants when the first is earlier than the second. Every Span entity can be assigned a temporal location. Temporal location is a primitive relation between an entity and a region of time. Instantaneous Span entities are located at instants of time. The co-temporal holding is a relation between Span entities which have as temporal locations exactly the same region of time. Each temporal part is thus the sum of all co-temporal parts of a Span entity located within a given region of time. Instantaneous temporal parts can be called temporal slices. By definition of a temporal slice, events are located at an instant of time. The relation “occurs at” is the relation which holds between a processual entity and an instant of time at which a temporal slice of this processual is located. The temporal regions of Span are summarized in Table 5.

Table 5. Temporal regions in Span ontology

Concept	Definitions
Time T	The constant time designates an individual which is the whole of time.
TimeRegion TR(x)	TR(x) means that x is a region of time, i.e. a part of time which may be extended or instantaneous, connected to various degrees or scattered. $\text{TimeRegion}(x) \equiv_{\text{def}} \text{Part}(x, \text{time})$
TimeInstant TI(x)	TI(x) means that x is an instant of time. TimeInstant is a specialization of the predicate TimeRegion. <i>Table 5 continues ...</i>

TemporalLocation $TL(x, t)$	<i>Table 5 continues ...</i> $TL(x, t)$ is a primitive relation between an entity and a region of time ' $TemporalLocation(a, t)$ ' stands for: ' a is located at region of time t '.
TemporalLocation at an Instant $TLI(x, t)$	$TLI(x, t)$ means that x is temporally located at t and that t is an instant of time.
TemporalCo-Location $TCoL(x, y)$	$TCoL(x, y)$ means that x and y are located at the same temporal region.
Temporal Subsumption $TSbL(x, y)$	$TSbL(x, y)$ means that x temporally subsumes y , i.e., the temporal location of y is a part of the temporal location of x .
TemporalPart $TP(x, y)$	Each temporal part is the sum of all cotermporal parts of a Span entity located within a given region of time. $TemporalPart(x, y) \equiv_{def} Part(x, y) \wedge \forall z ((Part(z, y) \wedge Cotermporal(x, z)) \rightarrow Part(z, x))$
TemporalSlice $TS(x, y)$	$TS(x, y)$ means that x is an instantaneous temporal part of y . $TemporalSlice(x, y) \equiv_{def} TemporalPart(x, y) \wedge \exists t AtTime(x, t)$ $AtTime(x, y) \equiv_{def} (TemporalLocation(x, y) \wedge TimeInstant(y))$
Event	The predicate Event is a holding of instantaneous processuals. $Event(x) \equiv_{def} \exists y (Processual(y) \wedge TemporalSlice(x, y))$
OccursAt	The relation OccursAt is the relation which holds between a processual entity and an instant of time at which a temporal slice of this processual is located. $OccursAt(x, t) \equiv_{def} \exists y (TemporalSlice(y, x) \wedge AtTime(y, t))$

7. Perdudant Entities Related to Cross-Cultural Meetings and Negotiations

In this section, we describe temporal entities related to cross-cultural meetings and negotiations between Japan and Finland. We aim to identify temporal regions according to the Span ontology.

Temporal concepts of cross-cultural meetings and negotiations have an effect on the dynamics of the whole meeting and negotiations process starting from the very first contact and ending up to possible collaborative actions. We have analyzed meeting and negotiation processes and identified fourteen conceptual classes which have temporal parts. Our analysis was based on experiences of fifteen Finnish companies working in Japan. We focused on the companies which are members of the Finnish Chamber of Commerce in Japan [5] and/or Finpro [6] because of their stable like position in Japanese business life. We used theme interview method with three main themes: life before, during and after a negotiation process and related meetings. The main themes were selected by means of the literary [4, 13, 15, 16] where process type approach to negotiations and meetings in Asia cultures was highlighted rather than just a single type event. The interviews were carried out by telephone discussions and/or by email. The interviewed persons represented middle management. The identified temporal parts and their subparts can be described as follows.

Class 1: Working time \supset {Holidays, Festivals, Business Hours, Academic Terms}: Major holidays, festivals, business hours and academic terms differ between Japan and Finland in some extent. The differences should be taken into account when scheduling collaborative actions, virtual and/or face-to-face meetings.

Class 2: Beginning of the Meeting: Meetings, be they face-to-face meetings or videoconferences, are not begun in the same way as we move from culture to culture. In Japan, the beginning phase takes 15–20 minutes including formal introduction, protocol seating, green tea, small talk, and then a signal from senior Japanese to begin. In Finland, the beginning phase takes 5 minutes including formal introduction, cup of coffee, sit down and begin.

Class 3: Structuring a Meeting \supset {Linear-Active, Multi-Active, Reactive}: Linear-active members need relatively little preamble or small talk before getting down to business. They like to introduce bullet points that can serve as an agenda. Tasks are segmented, discussed and dealt with one after one other, linearly in time. Solutions reached are summarized in the minutes. Finnish are quite linear-active people in meetings. Multi-active members prefer to take points in random order or in order of importance and discuss them for hours before listing bullet points as conclusions. Reactive people do not have the linear obsession with agenda, neither are they wooed by multi-active arguments. In Japan, for instance, things are not black or white, possible or impossible, right or wrong. They see arguments and ideas converging and ultimately merging in time. Japanese approach concentrates on harmonizing general principles prior to examining any details.

Class 4: Meeting Behavior \supset { Silence, Listening}: The deliberate use of silence can be an invaluable advantage in negotiations. In Finland and Japan, silence is not uncomfortable but is an integral part of social interaction. In both countries what is not said is regarded as important, and silence can last several minutes. Listening habits can also play an important part in the negotiating process. Both Finns and Japanese have a good ability to listen closely for long periods of time.

Class 5: Body Language \supset {Instantaneous Signs, Sequential Signs}: Finns and Japanese use body language that is well understood by fellow nationals. Their body language is very subtle. Both Finns and Japanese are accustomed to looking for minimal signs. The signs can be instant like, occasional signs or sequential signs with varying temporal intervals constituting a temporal entity.

Class 6: Objectives: For Japanese the current project or proposal is a trivial item in comparison with the momentous decision they have to make about whether or not to enter into lasting collaborative relationship with the foreigners. They seek long-term relationships. Usually, decisions have been made before the meeting by consensus. Japanese see meetings as an occasion for presenting decisions, not changing them. Finns appreciate long-term relationships and they can make decisions during the meetings.

Class 7: Professionalism: As far as professionalism is concerned, what is often forgotten is that negotiating teams rarely consist of professional or trained negotiators. Engineers, accountants and managers used to directing their own nationals are usually completely lacking in foreign experience. When confronted with a different mindset, they are not equipped to figure out the logic, intent and ethical stance of the other side and may waste time talking past each other.

Class 8: Social Setting: The Japanese regard a negotiation as a social ceremony to which important considerations of venue, participants, hospitality and protocol, timescale, courtesy of discussion and the ultimate significance of the session are

attached. They view the session as an occasion to ratify ceremonially decisions that have previously been reached by consensus. Finns have little feel for the social nuances displayed by the Japanese. In the straightforward egalitarian cultures, meetings are conducted without regards to social status. Although quite polite, Finns have difficulty in settling down to a role in meetings where social competence dominates technical know-how.

Class 9: Values: Finns rely on data, try to get as much action and decision making as possible into the hours available. They have a time/efficiency equation in their mind. The Japanese have their own aesthetic norms, which is bound up with a complex set of obligations. The Japanese see themselves as farsighted negotiators and courteous conversationalists.

Class 10: Compromise: It is not uncommon for negotiation to enter a difficult stage where the teams get bogged down or even find themselves in a deadlock. For Japanese compromising during a negotiation is a departure from the company-backed consensus. Adjournment is sometimes the only way out. Compromise may be defined as finding a middle course. The Japanese like to use go-betweens. Finns might rank directness above subtle diplomacy in discussions.

Class 11: Logic: In Japan, different views co-exist in the same space and time, without invoking conflict or urge to resolve differences. In Western logic, contradictory views of the same thing are not tolerated. Difference must be resolved. Singular and exclusive truths are looked for.

Class 12: Decision making \supset {Before the Meeting, During the Meeting, After the Meeting}: Negotiations lead to decisions. How long they take to be made will depend on the cultural groups involved. The Japanese prefer to let decision be made for them by gradually building up a weighty consensus instead of making decisions. A decision may take months. Once the Japanese have made their decisions they expect their partners to move fast toward implementation. What Finns fail to understand is that the Japanese, during the long decision making process, are simultaneously making preparations for the implementation of the project.

Class 13: Agreements: Finns regard a written agreement as something that is final. For the Japanese the contract is merely a statement of intent. They will adhere to it as best they can but will not feel bound by if conditions suddenly change.

Class 14: Relationship management \supset {Before the Meeting, In the Meeting, After the Meeting}: In Japan, before the meeting, it is good to arrange an introduction by a respected third party of high status, ideally someone known to both candidate collaborator and her/his Japanese counterpart. In Finland, it is good to begin by sending a letter, an email or a fax in English with basic information about the organization and the proposal, and follow up with a phone call. If the potential collaborator is interested, a date and time for a meeting can be confirmed. In the meeting, both Japanese and Finns give little gifts and after the meeting season greetings are appreciated.

Temporal parts related to cross-cultural meetings and negotiations and examples of temporal regions according to the Span ontology are illustrated in Figure 4. The overall concept of a perdurant entity, in this case in the form of temporal regions of the Span ontology, provides us an interesting framework for analyzing temporal entities related to cross-cultural meetings and negotiations as well as for modeling the progress of negotiation and meeting processes. Meetings and negotiations are social situations with a protocol including certain rules and forms. These rules and forms include some time-sensitive parts. In cross-cultural environments, there occur several protocols. These

protocols can collide or softly meet depending on the cultural competence of the participants. This concerns also the time-sensitive components of the protocols. For example, Finnish participants who are not aware of Japanese beginning phase of the meeting that takes around 15–20 minutes, would try to start the meeting at once according to their own protocol. In this case Finnish participants would cause negative tensions already at the very beginning of the meeting.

The added value of our approach could be summarized as follows:

- Time-sensitive components of cross-cultural meetings and negotiations provide interesting context to study the concept of perdurant entities.
- Identification of temporal entities of cross-cultural meetings and negotiations deepen our understanding how these social situations proceed.
- Ontology provides us a formal approach to describe temporal entities.
- Formal descriptions are the basis of designing and demonstrating different components of the CSIS system.

In practice, the graphical presentation of the temporal parts related to cross-cultural meetings and negotiations could be the basis of graphical interface for one of the educational components of the CSIS system. The component would illustrate the temporal progress of cross-cultural meetings and negotiations for project managers and team members. More detail information, examples of practical situations and best practices could be linked to the main topics illustrated in Figure 4.

8. Conclusions and Issues for Further Research

We have addressed three issues in our study. First, we have discussed implications of cross-cultural differences for meetings and negotiations. Second, we have presented three relevant ontological approaches – OWL-Time, temporal aggregates and temporal regions of the Span ontology – for identifying and describing temporal entities in the context of cross-cultural meetings and negotiations. We also aimed to discuss on the concept of the perdurant entity in the cross-cultural context. Third, we have given examples of applying those ontological approaches to scheduling meetings and recurrent actions over time zones and to describing temporal regions of cross-cultural meetings and negotiations between Finland and Japan.

Temporal entities have important role in design and implementation of culture-sensitive information systems (CSIS). Our study is exploratory and a small vertical part of the whole horizontal CSIS designing process. We have analyzed meeting and negotiation processes and identified fourteen conceptual main classes that are time-sensitive. Our analysis is based on interviewing Finnish companies working in Japan. In literature related to cross-cultural communication between Asian and Western cultures, use of time is discussed as one of the cultural dimensions. Literature that is focused on cross-cultural management issues highlights negotiations as processes. In our study we have identified temporal entities related to those processes. Our findings support the procedural approach reported in the literature to negotiations and related meetings in Asian countries.

The essential themes for further research are: (a) considering how to encode the temporal regions of Span ontology and identified temporal entities in OWL, (b) how the findings of our study could be utilized in designing cross-cultural XML Schema

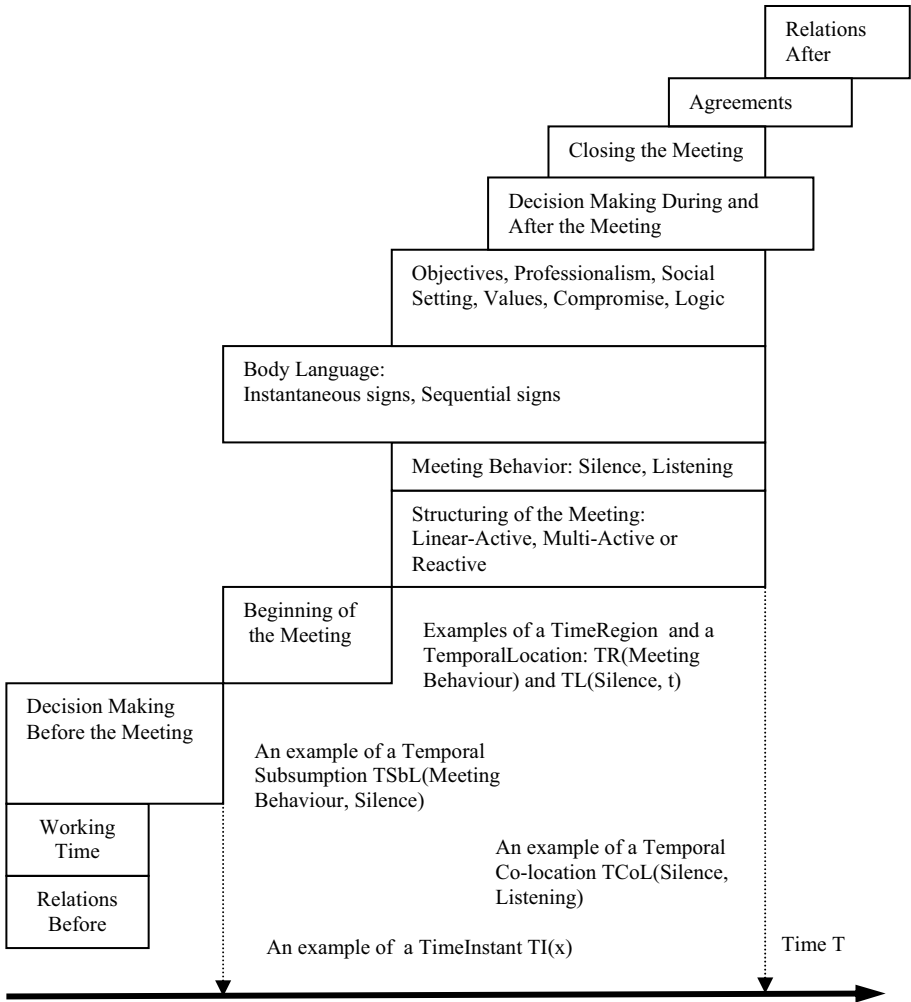


Figure 4. Temporal parts related to cross-cultural meetings and negotiations

modules needed in CSIS realization and (c) theoretical consideration of temporal dimensions; what if time is not a scalar but an n-dimensional vector.

By understanding different temporal dimensions and dynamics in cross-cultural meetings and negotiation processes we could improve our skills in cross-cultural communication and increase our cultural competence. Cross-cultural projects and teams that use CSIS systems could benefit from mutual learning experiences and innovative thinking to enhance the competitive position of their organizations. From operational point of view, cultural competence integrates and transforms knowledge about cultures, groups of people and individuals into specific practices and attitudes.

Acknowledgements

We express our deep thanks to the Scandinavia-Japan Sasakawa Foundation for funding our study and for EJC 2008 Programme Committee Reviewers for many constructive comments and ideas.

References

- [1] Allen, J. F. Time and Time Again: The Many Ways to Represent Time, *International Journal of Intelligent Systems* **6**, 4 (1991), 341–355.
- [2] Alonso, O., Gertz, M. and Baeza-Yates, R. On the Value of Temporal Information in Information Retrieval. *SIGIR Forum* (referred 25th Aug. 2008) <URL: http://sigir.org/forum/2007D/2007d_sigirforum_alonso.pdf>, 2007.
- [3] Bijl, A. *Ourselves and Computers. Differences in Minds and Machines*. London: Macmillan Press Ltd., 1995.
- [4] De Mente, B. *Etiquette. Guide to Japan*. Singapore: Tuttle Publishing, 2001.
- [5] FCCJ. Finnish Chamber of Commerce in Japan (referred 25th Aug. 2008) <URL: <http://www.fcc.or.jp/index-e.html>>, 2008.
- [6] Finpro. *Finpro* (referred 25th Aug. 2008) <URL <http://www.finpro.fi/en-US/Business/Market+Knowledge/Finpro+East+Asia/>>, 2008.
- [7] Hay, M. and Usunier, J.-C. Time and Strategic Action. *A Cross-Cultural View, Time & Society* **2**, 3 (1993), 313–333.
- [8] Heimbürger, A. When Cultures Meet – Modelling Cross-Cultural Knowledge Spaces. In: Jaakkola, H., Kiyoki, Y. and Tokuda, T. (eds.) *Proceedings of the 17th European – Japanese Conference on Information Modelling and Knowledge Bases*, June 4 - 8, 2007, Pori, Finland, 2007, 318–325.
- [9] Hobbs, J. R. and Pan, F. An Ontology of Time for the Semantic Web, *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing* **3**, 1 (2004), 66–85.
- [10] Hofstede, G. *Culture's Consequences, Comparing Values, Behaviors, Institutions, and Organizations Across Nations*. Thousand Oaks, CA: Sage Publications, 2001.
- [11] Hofstede, G. and Hofstede, G. J. *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and Its Importance for Survival*. New York: McGraw-Hill, 2004.
- [12] Hofstede, G. *Geert Hofstede™ Cultural Dimensions* (referred 25th Aug. 2008) <URL: <http://www.geert-hofstede.com/>>, 2003.
- [13] Holden, N. J. *Cross-Cultural Management. A Knowledge Management Perspective*. Harlow, UK: FT Prentice Hall, 2002.
- [14] King, W. R. A Research Agenda for the Relationships between Culture and Knowledge Management, *Knowledge and Process Management* **14**, 3 (2007), 226–236.
- [15] Lewis, R. D. *When Cultures Collide. Managing Successfully Across Cultures*. London: Nicholas Brealey Publishing, 1999.
- [16] March, R. M. *The Japanese Negotiator*. Tokyo: Kodansha International, 1990.
- [17] Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A. *WonderWeb Deliverable D18. Ontology Library* (referred 25th Aug. 2008) <URL: <http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf>>, 2003.
- [18] McGinnis, M. A. *Advanced Negotiations: Lessons from the International Arena*. *Proceedings of the 90th Annual International Supply Management Conference* May 2005.
- [19] Pan, F. A Temporal Aggregates Ontology in OWL for the Semantic Web. In *Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web*, Arlington, Virginia, 2005.
- [20] Smith, B. and Gren, P. *Basic Formal Ontology (BFO)* (referred 25th Aug. 2008) <URL: <http://www.ifomis.uni-saarland.de/bfo>>, 2007.
- [21] W3C. *Time Ontology in OWL, W3C Working Draft 27 September 2006* (referred 25th Aug. 2008) <URL: <http://www.w3.org/TR/owl-time/>>, 2006.

A Common Framework for Board Games and Argumentation Games

Jenny ERIKSSON LUNDSTRÖM^{a,1}, Jørgen FISCHER NILSSON^b and
Andreas HAMFELT^a

^a*Department of Information Science, Computer and Systems Science Division, Uppsala University, Sweden*

^b*DTU Informatics, The Technical University of Denmark, Denmark*

Abstract. In this paper we discuss models of formal argumentation games. We argue that the tactics and strategies of board-games like chess provide a useful analogy for adversarial argumentation games. The objective of this study is to elaborate on a common model for board games and argumentation dialogues. In particular we strive at making analogies between tactical and strategic chess game notions and notions in adversarial argumentation games.

Keywords. Argumentation Games, Board Games, Metalogic, Game Trees

1. Introduction

In previous papers [1,2,3] we have suggested modelling (adversarial) argumentation as a two-party zero-sum game [4] with speech acts replacing moves. In this paper we proceed with this approach aiming at setting up a general model of two-agent games subsuming board games of chess as well as argumentation games. More specifically we try to establish an analogy between tactical and strategic notions of chess and similar notions in argumentation games.

Our approach rests on a metalogic framework in which board games are formulated in an appropriate logic and moves are conceived of as speech acts using this logic. This means that the argumentation process is modelled in a metalogic setting with background knowledge and speech acts forming object language clauses encoded as terms. The nondeterministic nature of the argumentation process is then conveniently captured as a metalogic program managing the speech act sequence forming the dialogue.

Although we have in mind primarily human dialogues our model is also to support agent dialogues, protocols and negotiations in multiagent systems.

The paper is organized as follows: In Section 2 we give a brief introduction to algorithmic analysis of games using game trees. In the following Section 3 we discuss board game set up, and in Section 4 we present a logical specification of the game analysis as discussed in the previous sections. In Section 5 we extend the logical specification to

¹Corresponding Author: Jenny Eriksson Lundström, Department of Information Science, Computer and Systems Science Division, Uppsala University, Box 513, SE-751 20 UPPSALA, Sweden; E-mail: jenny.eriksson@dis.uu.se.

model game states. In Section 6 we show in more detail our argumentation model, and in Section 7 we discuss the similarities of board games and argumentation games. In Section 8 and 9 we elaborate on the analogy between board games and argumentation. We conclude the paper in Sections 10 and 11 with a discussion of related work and possible research extensions.

2. Game Trees

Games are analyzed algorithmically in artificial intelligence using the notion of game trees. A node in a game tree represents a state or situation in a game and the immediate subnodes of a node represent admissible subsequent states after a party has made a move. We call the first agent the proponent and the second agent the opponent, as it is usual in *e.g.* legal proceedings.

Table 1. Game tree analysis

	won	lost
pro	<u>some</u> act leads to lost for the opponent (at turntaking)	<u>all</u> acts lead to won for the opponent (at turntaking)
opp	<u>some</u> act leads to lost for the proponent (at turntaking)	<u>all</u> acts lead to won for the proponent (at turntaking)

Depending on the nature of the game, game trees may make it possible to determine whether a certain game state is won or lost according to the specification in Table 1. One observes that the notion of won and lost are mutually recursively defined. The notion of won is associated with existential quantification whilst the notion of lost is associated with universal quantification. This table falls short of explaining cases where the state is neither won nor lost. Moreover, the table also has to account for analysis of leaf nodes in the game tree corresponding to states where the game is completed. In the table we use the term ‘act’ instead of ‘move’ preparing for games where the acts are *e.g.* speech acts like in dialogue games.

Before discussing dialogue games in the following section, we discuss board games. Traditionally board games involve movement of physical objects on a board, like in chess or tic tac toe. However, at the level of information analysis and modelling, the movement of physical objects are described as speech acts put forward according to protocols for the game. For instance for chess there are game description systems as discussed below.

3. Board Games Exemplified by Chess

In chess the prime completion criterion for a game is mate and for tic tac toe it is three in a row. For chess this is specified in the following Table 2. as extension of Table 1., where according to convention the proponent is called white and the opponent is called black.

In Table 3. we have included a column accounting for the draw situation in chess.

Table 2. Won and lost chess games

	won	lost
white		mate for white
	<u>some</u> act leads to lost for black (at turntaking)	<u>all</u> acts lead to won for black (at turntaking)
black		mate for black
	<u>some</u> act leads to lost for white (at turntaking)	<u>all</u> acts lead to won for white (at turntaking)

Table 3. Won, lost and draw chess games

	won	lost	draw
white		mate for white	no act and not mate (stalemate)
	<u>some</u> act leads to lost for black (at turntaking)	<u>all</u> acts lead to won for black (at turntaking)	
black		mate for black	no act and not mate (stalemate)
	<u>some</u> act leads to lost for white (at turntaking)	<u>all</u> acts lead to won for white (at turntaking)	

A completed game consists of a succession of completed states ending in a won or lost state. However, for board games there may also be a form of draw manifest as an infinite path through the game tree. This is exemplified in chess by leaving only the two kings at the table to move infinitely. No such equivalent is available in tic tac toe. The case of an infinite path is not specified explicitly in the tables.

4. Metalogical Setup

We now intend to formulate a logical specification of the game analysis outlined in the previous sections.

4.1. Logical Specification of Won and Lost

In all the logical specifications we assume available a predicate *move*(*S*₁, *S*₂) which provides all the successor states of a game state *S*₁. So for a given *S*₁ there are 0 or more successor states *S*₂. A specific game proceeding to a state *S*_{*n*} can be specified as a sequence *S*₁, *S*₂, ..., *S*_{*n*} where *S*_{*i*+1} is an admissible move in state *S*_{*i*} in accordance to the *move* predicate.

We cannot accept moves that are inconsistent with the state of the game. The admissible moves are all the moves that are consistent with the rules of the game and the state of the game. Thus the predicate *move* is a state transformer.

We formalize won and lost by two mutually recursively defined predicates *won* and *lost* cf. Table 2.

$$won(S_1) \leftarrow move(S_1, S_2) \wedge lost(S_2)$$

$$lost(S_1) \leftarrow mate(S_1)$$

$$lost(S_1) \leftarrow \forall S_2 (move(S_1, S_2) \rightarrow won(S_2))$$

The first clause tells that a state S_1 is won if there is an admissible move leading to a next state S_2 which is lost. Whereas S_1 is universally quantified S_2 can be understood as existentially quantified as it only appears in the antecedent of the implication.

Won and lost can now be specified formally as logic programming clauses where the universal quantification for *lost* is managed by negation as failure by way of the following rewriting of the third clause above:

$$lost(S_1) \leftarrow \neg \exists S_2 (move(S_1, S_2) \wedge \neg won(S_2))$$

In case of infinite game trees this program is not necessarily terminating unless a depth bound is included.

When the game states S_1 and S_2 are formulated as sets of logical sentences we have to take resort to metalogic and metalogic programming. This means that the logical sentences of the states have to be encoded as terms in the above logic program. For a further account of this mechanism we refer to our [5,3,2,1].

5. Logical Specification of Game States

In the logical specification we choose to model a game state by means of a finite set of sentences in an appropriate logic. Thereby we strive to achieve a generalized formal model of games which also covers dialogues for negotiation and argumentation to be elaborated in subsequent sections.

5.1. Logical Setup for Chess

A state of the board in a chess game is basically specified by listing the positions of pieces present at the board.² The presence of a piece can straightforwardly be modelled say by a predicate

$$piece(\langle colour \rangle, \langle type \rangle, \langle column \rangle, \langle row \rangle)$$

Moves are state changes depending on the rules of the board game. In tic tac toe an admissible move is simply putting a piece in an empty place at the board. In chess an ordinary admissible move is retracting a piece from the board and putting that piece in another position, possibly removing the opponent's piece if present at the new location. With this setup one may elaborate a logical specification of the rules of chess defining admissible moves in a given state. A specification of chess as definite clauses is given in [6]. For instance the presence of the white king at e1 is then stated as *piece(white, king, e, 1)*. Moving of the king from e1 to f2 is then modelled by removing the above atomic sentence from the state and replace it with *piece(white, king, f, 2)*.

²Some additional information may be needed e.g. whether a rooking has taken place.

6. Our Argumentation Game Setup

This section describes in more detail the argumentation model that we have presented in [1,2,3].

The key component in our argumentation games is a public common knowledge base being steadily extended during the argumentation with the speech acts. The knowledge base contains in general indisputable and defeasible sentences in propositional defeasible logic *e.g.*, as in Governatori *et al.* [7]. We can make a distinction between indisputable sentences representing indisputable assertions and defeasible sentences expressing assertions, which can be overruled by additional information and in particular by indisputable sentences. The defeasible sentences accommodate claims subject to disputes unlike the usual indisputable sentences which cannot be disputed. By itself this does not imply that we are using defeasible logic as a proper belief logic, since in defeasible logic in its basic form, the assertions do not contain nested belief modalities.

Initially this knowledge base is like the initial situation of a board game comprising jointly accepted knowledge, say rules of the game or clauses in statutory law. In addition, each of the parties have a repository of their own comprising the sentences that they can put forward in the course of the argumentation. This means that the game is closed in the sense that we in the form discussed here do not consider adducing new sentences in the course of the argumentation game.

Each party may internally entertain contradictory propositions. However, we require the speech acts put forward by either party to be *defeasibly consistent* meaning consistent in the sense that a party cannot contradict herself, not even defeasibly. This means of course that the uttering of some speech act may prevent the utterer from uttering some other speech act later on in a particular dialogue. In general this ensures the dynamical argumentation dialogue from being reducible to blackboard models of argumentation where everything is visible and in effect from the start.

A certain state in the argumentation game is like a state in a board game. In a state in the argumentation game there are certain admissible speech acts in analogy to the admissible moves in the board game. The two parties alternately put forward sentences from their respective repository being pooled into the common knowledge base in a turn-taking process until a terminal state is reached as to be specified below.

Definition 1 (argumentation game) *An argumentation game is constituted by a quintuple $(kb, rep_{prop}, rep_{opp}, keyclaim, wrongorder)$.*

A specific argumentation game is determined by the following components:

A common knowledge base of sentences kb , repositories of each of the parties Rep_{Party} , a key claim κ for the proponent and a binary relation $wrongorder$ imposing commonsense restrictions on the ordering of utterances. The binary relation $wrongorder$ is empty if there are no restrictions on the ordering. The first three components are finite sets of clauses in the defeasible logic and $keyclaim$ κ is a single clause. In addition a repository contains the distinguished atomic clause, *resting*, which is a speech act meaning that the party rests. The resting clause is always implicitly present. For our framework we assume that the common knowledge base remains logically consistent, albeit usually not defeasibly consistent by the very nature of disputing.

In the present framework we wish to stress the computational dialectics of argumentation. This implies that we consider a dialogue to be a process where the two parties take turns in putting forward utterances, propositions in defence of their respective claims.

Definition 2 (dialogue) *A dialogue is a speech act sequence with alternating contributions of the parties of a particular argumentation game.*

Definition 3 (completed dialogue) *A completed dialogue is constituted by a whole speech act sequence ending in a terminal state. A state is a terminal state if the game is won, lost or draw.*

7. Board Games vs. Argumentation Games

Consider argumentation games or disputes between two parties called the proponent and the opponent. The two parties put forward and exchange utterances (speech acts) according to game rules or certain protocol schemes. In the simplest case, the speech acts consist of propositions expressing facts, statutes, evidence etc., stated in an appropriate logic. There should be available also means of deducing logical consequences of the claims together with the available background knowledge.

In an argumentation game, if one party claims p and the other party claims $\neg p$ these claims together constitute an unacceptable contradiction. Therefore in formal argumentation one often resorts to defeasible reasoning, where propositions can be defeated by other propositions without causing logical contradictions. Defeasible propositions are therefore often used for expressing assertions which can be rejected by classical non-defeasible propositions representing indisputable facts [8,7,1]. This is to be distinguished from the fact that in argumentation games usually utterances are irrevocable and thus not to be withdrawn. In board games the movement of a piece may imply nonmonotonic update of the knowledgebase.

In board games as discussed above the parties aim at achieving a winning constellation at the board; in chess this is called mate. In argumentation games the parties strive at arguing convincingly in favor of their respective claims. Often the proponent is to argue in favor of the distinguished keyclaim which is then to be refuted by the opponent. In board games the effect of the pieces is to prevent certain constellations of the other party and at the same time promote a support of other pieces. Analogously in argumentation games utterances are put forward in order to refute or support the keyclaim. It is a general requirement here that the parties do not create contradictions by putting forward propositions which are inconsistent with previous propositions or with rules of the game. This means of course that the uttering of some speech act may prevent the utterer from uttering some other speech act later on in a particular dialogue. This is akin to presence of a piece in a position preventing a party from making certain moves. Thus, in argumentation games as well as in chess games the parties should all the time reflect on how a move will effect future moves. Game tree exploration is an algorithmic process for performing this deliberation.

8. Analogy Between Board Games and Argumentation

In the following tables we seek to establish an analogy between chess notions in the first column and argumentation dialogues in the second column.

Table 4. Analogous chess notions and argumentation notions

general chess notions	argumentation notions
piece moved to position on the board	defeasible proposition put forward as speech act
move of piece	defeasible proposition retracted and a new one put forward
capture of piece	defeasible proposition rebutted
support/defence of piece	proposition logically supporting other propositions
effect of piece	deductive closure of proposition
threatening effect of piece	threat of proposition in effect
rules and constraints for pieces	constraint for speech acts <i>e.g.</i> consistency

Furthermore, in the analogy we are trying to establish between board games and argumentation games we now consider more abstract concepts. We elaborate on the notion of threat. Here we consider threat of an argument to another analogously as the immediate threat posed by one piece to another:

Table 5. Chess and argumentation notions for threatening moves

specific chess notions	argumentation notions
quiet move	a speech act neither posing a threat nor effecting the balance
threat	move effecting the balance
attack	immediate threat of adversary’s argument
undermining	undercutting of adversary’s argument by attack on its premise
refute of <i>p</i>	defeater argument, blocking the conclusion <i>p</i> but not supporting $\neg p$
counterattack	immediate threat to adversary’s attacking argument

In the below Section 9. we augment this comparison with chess notions involving combinations of chess moves.

8.1. Termination Criteria

The challenge of the dialogue or argumentation game is to conduct a winning strategy in analogy to the accomplishment or elaboration of a winning strategy in two-player games. More precisely, the proponent is to justify or at least defend a key claim, whereas the intent for the opponent is to dispute or preferably defeat this key claim as to be detailed below. Parts of this section survey our won and lost analysis from [2].

Our analysis refers to an argumentation setting where usually the opponent’s claim is granted by the proponent’s failure to prove the keyclaim κ as it appears in the Table 6. This is common in *e.g.* legal argumentation settings. Thus, *won* and *lost* are mutually recursively defined in the underlying game-tree exploration as follows:

Table 6. Analogous chess notions and argumentation notions

chess notions	argumentation notions
capture	irrefutable rejection of captured argument
check	a proposition which attack though not defeat the keyclaim
check mate	refutation of the adversaries keyclaim

Table 7. Won and lost dialogues from our [1,2]

	won	lost
pro	$S_i \vdash \kappa$	$S_i \vdash \neg\kappa$
	$S_i \not\vdash \kappa \wedge \text{opp-rep exhausted in } S_{i+1}$	$S_i \not\vdash \kappa \wedge \text{prop-rep exhausted in } S_{i+1}$
	<u>some</u> act leads to lost for the opponent (at turntaking)	<u>all</u> acts lead to won for the opponent (at turntaking)
opp	$S_i \vdash \neg\kappa$	$S_i \vdash \kappa$
	$S_i \not\vdash \neg\kappa \wedge \text{prop-rep exhausted in } S_{i+1}$	$S_i \not\vdash \kappa \wedge \text{opp-rep exhausted in } S_{i+1}$
	<u>some</u> act leads to lost for the proponent (at turntaking)	<u>all</u> acts lead to won for the proponent (at turntaking)

Table 7. tells that the game is lost for the proponent if the negation of the key claim is affirmed strictly or if the game is completed and the keyclaim does not follow even defeasibly. The symbol \vdash represents strict or classical proof, while the symbol $\not\vdash$ represents defeasible proof. A classical proof of the proof of the proposition is also a defeasible proof of the proposition but not vice versa. As in *check mate* in the table 6. we consider the game as *won* for the proponent (Pro) and *lost* for the opponent (Opp) if the keyclaim κ is definitely proven in the state S_i , $S_i \vdash \kappa$. In order to accommodate the dynamic character of our approach, an argumentation state is also *won* for the proponent and *lost* for the opponent if the keyclaim is defeasibly proven in state S_i and in the subsequent state S_{i+1} the opponent has exhausted all his moves, $S_i \not\vdash \kappa \wedge \text{opp-rep exhausted in } S_{i+1}$. The last criteria of *won* and *lost* for a party define the exploration of the game tree using existential and universal quantification alternately (cf. the minimax principle for exploring game trees [4]). Thus, the game is *won* for a party in the state S_i if some admissible act leads to *lost* for the other party in a subsequent state S_{i+1} . Accordingly, the game is *lost* for a party when all admissible acts recursively lead to won for the other party. Equivalently to *check mate* and the first *won* criteria for the opponent, the game is *lost* for the proponent when the negation of the keyclaim is definitely provable in the state S_i , $S_i \vdash \neg\kappa$.

As stated above, for some real-life domains, the opponent's claim is granted merely by the non-provability of the proponent's keyclaim rather than its classical negation. Thus, the second *lost* criteria for the proponent, $S_i \not\vdash \kappa \wedge \text{prop-rep exhausted in } S_{i+1}$, requires that in the state S_{i+1} the proponent has exhausted all her admissible moves and the keyclaim cannot be defeasibly proven in the given state S_i . This should be contrasted to the second *won* criteria for the opponent, $S_i \not\vdash \neg\kappa \wedge \text{prop-rep exhausted in } S_{i+1}$, which

states that the game is won for the opponent when the *negation* of the keyclaim has been defeasibly proven in the given state S_i .

Definition 4 (draw) *Analogous to the stalemate, a draw is when the the game is neither won nor lost for either party.*

In many cases a dialogue game may be expected to come out as a draw. That is, as analogous to a *stalemate* in chess, the proponent can defend her proposition against attack, but on the other hand she lacks admissible arguments/moves to justify it by logical means given the available evidence and presented counter-arguments. However, human interaction is governed by many higher level principles incorporating institutional (societal) values on the dispute resolution cf.[9]. Thus it should be noted that in many semiformal real-life domains *e.g.* legal settings, the draw situation is not modelled as a balanced situation. Instead it is solved by placing a burden of proof on either party in accordance to the norms or higher level principles governing that particular interaction. *e.g.* in a criminal litigation, in case of doubt, *the prosecutor* has to convince/prove that the accused is guilty beyond a reasonable doubt, while *the accused* only has to produce an exception to the accusation.

From the point of view of argumentation games the keyclaim in chess effectively says that the white king is persistently defensible and the black king is not persistently defensible, in other words that white can mate black. White is not losing if this keyclaim is not proven.

8.2. Updating the Knowledge Base States

We have to distinguish monotonic and nonmonotonic updates in the course of the argumentation game.

Definition 5 (move_{monotonicextension}) *In a monotonic extension of the common knowledge base a move is constituted by the putting forward of a speech act, in resemblance to putting a chess-piece on a position on the board.*

In a monotonic extension of the common knowledge base a *move* is constituted by the putting forward of a speech act, in resemblance to putting a chess-piece on a position on the board. Uttered speech acts are never retracted from the knowledge base during the dialogue, only committed advancement of sentences are allowed for realistic argumentation reasons.

The repositories delineate possible future utterances at the current state for the purpose of the game analysis. Thus, in the course of the dialogue the knowledge base is monotonically extended whereas the repositories are dynamically retracted.

Alternatively, we could consider a *move* to retract uttered speech acts, in analogy to moving a piece to another position at the board. As such our knowledge base bears resemblance to the notion of epistemic states in Gärdenfors [10] and belief revision.

Definition 6 (move_{beliefrevision}) *In a non-monotonic extension of the common knowledge base a move is constituted by a removal of speech acts and the putting forward of another speech act.*

9. Tactics in Argumentation Games - Combinations of Moves

In this section we define some more advanced chess notions in relation to argumentation games. In most cases these notions are referring to sequences of moves.

Definition 7 (combination) *A combination is a sequence of arguments/moves that together force the adversary to put forward a specific counterargument that will leave the adversary with very few possible lines of continuation.*

In a dialectical dispute a party may refrain from putting forward utterances *prima facie* in support of her claim. This is because utterances may *backfire* during the course of the dispute, say by providing evidence that happens to be in favor of the other party.

Definition 8 (pin) *A pin is when a party cannot present an admissible argument because it would backfire and become a supporting argument for the adversary.*

Definition 9 (absolute pin) *An absolute pin is a pin against the keyclaim.*

Definition 10 (gambit) *A gambit is a sacrifice by corroborating some sub-argument of the adversary in order to support the own claim.*

Sometimes a single proposition may play the role of a combination of moves:

Definition 11 (fork) *A fork is the putting forward of a complex question, i.e. a proposition implying a double attack which cannot be countered with one admissible speech act.*

Definition 12 (zwischenzug) *A zwischenzug is the putting forward of a proposition that forces a response from the other party that weakens his position.*

10. Related Work

Substantial work in the field of computational dialectics has proven argumentation games to be a promising candidate for modelling legal reasoning (see e.g. [11], [12],[13], [14]) For a comprehensive survey see [15]. Elaborating on the notion of support Dung [16], various forms of argumentation semantics have been proposed in order to capture different aspects of argumentation (for a discussion see e.g. [17]).

An early specification and implementation of a dialogue game based on the Toulmin argument-schema is presented in [11]. The framework is based on the Toulmin argument-schema without a specified underlying logic. Focusing on identifying the issues in the argumentation, a fully implemented computational model of argumentation called The Pleadings Game is presented in [12]. However, the goal of the model was to identify issues in the argumentation rather than as in our case elaborating on the status of the main claim. DiaLaw [13] is a model of a two-player argumentation game in which rhetorical as well as psychological issues of argumentation are highlighted. In contrast to our work, the main focus for the two players is to convince each other rather than defeating the adversary. Vreeswijk [18] provides abstract argumentation systems in which he uses metagames for changing the rules. Also by use of metalogic programs, Dung et.

al [19] present dialectic proof procedures for argumentation. Still, this approach is built on assumptions that are atomic, whereas in our framework the arguments are expressible as rules of propositional defeasible logic represented as terms in the metalanguage. Other significant contributions on modes of argumentation games have been made by Prakken and Sartor. [14] proposes *Augmented Litigation inference system* (ALIS) which distributes the burden of prosecution as a result of an argument based reasoning. Although inspired by this work, we emphasize the dynamics of the dialogue and focus on providing a general approach to modelling argumentation. In our model the knowledge of a party is partitioned into private knowledge and common knowledge. The private knowledge, which is unknown to the other party, becomes part of the common knowledge by a party putting forward speech acts from its private knowledge base. Thus a main difference, enabling the analogy to games, in our framework the parties in general advance their available knowledge and claims stepwise. In addition, also specific to our approach is the use of game tree exploration, a special case of min-max tree analysis, from artificial intelligence [4].

11. Conclusion

In this paper we have presented a common formal framework for board games and argumentation games focusing on chess games. We would like to extend this study to cover other board games like *go*, which are substantially different from chess. We are also going to study possible connections to Wittgenstein's language games [20] and Lorenzen style dialogue games [21].

References

- [1] A. Hamfelt, J. Eriksson Lundström, and J. Fischer Nilsson. A metalogic formalization of legal argumentation as game trees with defeasible reasoning. In *ICAIL'05, Int. Conference on AI and Law, University of Bologna CIRSFID, Bologna, Italy June 6-11 2005*, pages 84–89, 2005.
- [2] J. Eriksson Lundström, A. Hamfelt, and J. Fischer Nilsson. Legal rules and argumentation in a metalogic framework. In *JURIX 2007, Int. Conference on AI and Law, Leiden, The Netherlands, December 12-15*, pages 39–48, 2007. IOS Press.
- [3] J. Eriksson Lundström, A. Hamfelt, and J. Fischer Nilsson. A rule-sceptic characterization of acceptable legal arguments. In *ICAIL 2007 Stanford University Stanford, California, USA, June 4-8*, pages 283–284, 2007. ACM Press.
- [4] N. J. Nilsson. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann California, 1998.
- [5] A. Hamfelt. and J. Fischer Nilsson. Towards a logic programming methodology based on higher-order predicates. *New Generation Computing*, 15, pages 421–448, 1997.
- [6] M. Genesereth, N. Love, and B. Pell. General game playing: Overview of the AAAI competition. *AI Magazine*, pages 63–72, Summer 2005.
- [7] G. Governatori, G. Antoniou, D. Billington, and M. J. Maher. Argumentation semantics for defeasible logics. *Journal of Logic and Computation*, 14(5), pages 675–702, 2004.
- [8] D. Nute. *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3 Defeasible logic, pages 353–395. Oxford University Press, 1994.
- [9] Andreas Hamfelt. Formalizing multiple interpretation of legal knowledge. *Journal of Artificial Intelligence and Law*, 3(4), pages 221–265, 1995-1996.
- [10] P. Gärdenfors. *Knowledge in Flux - Modeling the Dynamics of Epistemic States*. 1988.
- [11] T. J. M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In *Proceedings of JURIX 98 Nijmegen*, pages 5–20, 1984. JURIX, GNI.

- [12] T. Gordon. The Pleadings Game: An artificial intelligence model of procedural justice. *Journal of Artificial Intelligence and Law*, 2(4), pages 239–292 1993.
- [13] A. R. Lodder. Dialaw: On legal justification and dialogical models of argumentation. *Artificial Intelligence and Law*, 8(2/3) pages 265–276, 2000.
- [14] H. Prakken and G. Sartor. Formalising arguments about the burden of persuasion. In *ICAIL 2007*, pages 97–106. ACM Press, 2007.
- [15] C. I. Chesñevar, A. G. Maguitman, and R. P. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4), pages 337–383 2000.
- [16] P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning and logic programming and n -person game. *Artificial intelligence*, 77 pages 321–357, 1993.
- [17] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127, pages 187–219, 2001.
- [18] G. Vreeswijk. Representation of formal dispute with a standing order. *J. Artificial Intelligence and law*, 8(2-3) pages 203–230, 2000.
- [19] P. M. Dung, R. A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based admissible argumentation. *Artificial Intelligence*, 170 pages 114–159, 2006.
- [20] L. J. J. Wittgenstein. *Philosophical Investigations*. Blackwell Publishing, 1953-2001.
- [21] K. Lorenz and P. Lorenzen. *Dialogische Logik*. Darmstad, 1978.

The Problem of Tacit Knowledge – Is It Possible to Externalize Tacit Knowledge?

Ilkka VIRTANEN

Department of Computer Sciences, University of Tampere

33014 University of Tampere, Finland

ilkka.virtanen@cs.uta.fi

Abstract. Various authors in the field of knowledge management have adopted the view that individuals' tacit knowledge should be externalized and shared in organizations. According to Polanyi's original theory of tacit knowing, an explicit expression of tacit knowledge is, however, considered very difficult, even impossible. We studied this contradiction by analyzing Polanyi's theory of tacit knowing in order to consider the correctness of the idea of externalization of tacit knowledge. Despite the fact that tacit knowing is an essential basis for all knowledge, we claim that tacit knowledge cannot be externalized in a way it is presented in the knowledge management literature.

Keywords: tacit knowledge, explicit knowledge, focal awareness, subsidiary awareness

Introduction

Tacit knowledge has been a popular concept since early 90's in the area of knowledge management. The origin of the concept is traced back to Polanyi's philosophy of knowing [19]. The basis of Polanyi's theory is the observation that "we can know more than we can tell." [21, p. 4] Thus, tacit knowledge refers to an individual knowledge that is highly personal and hard, even impossible to express or share with others.

The main motivation for the popularity of the concept in knowledge management discussion is the widely supported claim that organizations can achieve competitive advantages by using effectively their unique knowledge [e.g. 6, 23]. According to many authors, tacit knowledge is an important source of unique knowledge [e.g. 1, 2, 9]. These theories suggest in many cases that individuals' tacit knowledge should be externalized and shared with other members of the organization. According to Nonaka and Takeuchi [14, p. 64], externalization is a process of articulating tacit knowledge into explicit concepts. This kind of process aims to creation of new knowledge that may lead to innovations and more efficient internal functions of the organization.

Polanyi's theory illustrates that knowledge possessed by humans is more complex phenomenon than often considered. This is a fundamental, yet sometimes ignored, basis also in information system science; as Kangassalo [8] remarks, although the idea that shared knowledge in information systems forms a globally understood

phenomenon sounds good, it contains various difficulties in practice. One of those problems discussed here is that information systems do not guarantee objective understanding of knowledge, because language cannot fully translate persons' inner representations of it. This question is naturally even more challenging in multilingual environments. Moreover, according to Hori and Ohsuga [7], we cannot do anything with computers unless some structures of the real world are mapped onto a representation in the computer. They suggest that it cannot be assumed that some structure of the real world can be captured rationally and unambiguously. They call this problem of knowledge representation *an articulation problem*. They see it as a "core-problem" of information modelling. According to Tuomi [25, p. 113], there has been generally too little emphasis on the sense-making aspects of information systems. Sense-making refers to creation of situational awareness and understanding of complex phenomena. Klein *et al.* [10, p. 71] characterize sense-making as "a motivated, continuous effort to understand connections (which can be among people, places, and events) in order to anticipate their trajectories and act effectively." For example, in conceptual modelling it is important to understand how tacit knowledge affects the way humans explain and structure concepts and relations between them – no modeller can rely only on the formal knowledge of the target. These questions should be faced before planning of a system. This is why tacit knowledge and personal understanding can be considered as an increasingly important question also in information system science.

According to Polanyi's original definition of tacit knowledge it cannot be expressed explicitly. This contradiction becomes obvious despite the fact that knowledge management theories are claimed to be built on Polanyi's thinking. The question is important in current knowledge management discussion; if externalization of tacit knowledge is considered impossible, the application of the concept in certain contemporary theories can be seen very questionable, even wrong. On the other hand, it may also be possible that the mentioned theories are practical extensions of Polanyi's theory, meaning that they do not contradict with Polanyi's thinking. The difference between these options is significant.

We argue that an essential problem is that the concept lacks a commonly agreed definition; too little attention has been addressed to the question, *what tacit knowledge really is?* The question is often bypassed by remarking that tacit knowledge is the kind of knowledge that is hard to articulate – this obviously can mean many things. Based on Polanyi's original theory of tacit knowing, we claim that the idea of externalization of tacit knowledge indeed seems too simplified in certain respects. It should be, however, taken into consideration that we do not question the idea of knowledge creation and its methods, but restrict our analysis to the use of the concept of tacit knowledge.

We first introduce Polanyi's theory by presenting its epistemological motivation and then concentrating on the part concerning tacit knowing. Based on Polanyi's theory we then reconsider how accurately Polanyi's thinking is taken into consideration in the contemporary knowledge management literature.

1. Related Work

The question about the exploitation of tacit knowledge arose in the mid 90's. The most important impulse to contribute to the generalization of such a view was given by the

theory of organizational knowledge creation by Nonaka and Takeuchi in 1995 [14]. They presented a so-called SECI-model that explains a creation of new knowledge by means of conversions between tacit and explicit knowledge. The most essential part of the theory is an operation where tacit knowledge is converted to explicit knowledge. Nonaka and Takeuchi call this operation *externalization*. They point out that tacit knowledge is hard to articulate and must therefore be externalized indirectly by an illustrative use of language: for example by using metaphors, analogies and stories person's tacit knowledge can be shared and taken advantage of by other persons. According to Tsoukas [24], ever since the publication of Nonaka and Takeuchi's theory it has been nearly impossible to find a publication on knowledge management that does not make a reference to or use the term "tacit knowledge". Many authors have adopted Nonaka and Takeuchi's view supporting the idea of exploitation of tacit knowledge [e.g. 1, 2, 9]. It seems reasonable to state that their theory has influenced significantly ways of thinking in knowledge management.

Some authors, however, criticize Nonaka and Takeuchi's view. Cook and Brown [4] claim that explicit and tacit knowledge are different forms of knowledge, and conversion from one form to another is impossible; neither of the forms can substitute the other nor be its variant.

According to Orlikowski [15], tacit knowing is an inseparable part of action because it has developed in action. Consequently, tacit knowledge is bound to practice and maintains that form. Thus, it cannot be shared.

Tsoukas [24] argues that Nonaka and Takeuchi's view is erroneous because they have ignored the ineffability of tacit knowledge. He claims that tacit knowing cannot be captured or translated but is only manifested in what humans do. People can learn from others through social interaction, but it does not mean that something tacit becomes explicit.

Tsoukas and Orlikowski seem to stress knowing in practice instead of knowledge itself. According to that view, tacit knowledge cannot be transferred due to the inseparability between knowledge and action. Nonaka and Takeuchi seem to treat knowledge as something more objective, because they see externalization and transfer of tacit knowledge possible. The difference between these views is important. It raises a need to clarify what Polanyi's epistemological basis is for the concept of tacit knowledge.

2. Polanyi's Theory of Knowing

Polanyi's cognitive theory developed during over three decades in various writings. He believed that modern epistemological theories had described human knowledge far too narrowly since the birth of western philosophy, because an absolute objectivity was emphasized as an attainable ideal for knowledge; knowledge was seen as something detachable and independent of knower. According to Mitchell [13], for Polanyi 'objectivism' was shorthand for a collection of epistemological theories rooted in Descartes' and Locke's thinking. In Descartes' [5] thinking human was able to produce wholly objective knowledge of surrounding world. Descartes further assumed that anything not recognized by human reason could not be acknowledged as knowledge; knowledge had to be something distinctive and verified. Also, in the tradition of empirism a pursuit of common and objective laws that described phenomena of the

reality was seen ideal (for example Locke [12]). This view refers to positivist philosophy in a sense that knowledge is seen as an entity that can exist independent on the knower, for example in a written form.

Polanyi saw theories basing on objectivistic traditions wrong, even destructive, for personal participation was included in every act of knowing in his thinking; even scientific discoveries were often made based on unexplained informed guesses, intuitions and imaginative ideas that reflected some kind of *hidden knowledge*.

Polanyi [21] claims that objectivistic theories had ignored this *tacit dimension* of knowledge. The tacit dimension means briefly that every piece of knowledge has knower-dependent, tacit elements, on which the explicit dimension of the knowledge is built. Personal meaning and understanding of knowledge are based on this process; mind does not reflect reality in a passive way, but actively tries to build up the understanding of it.

In Polanyi's thinking there has to be a knowing subject for the existence of any knowledge. The idea refers to constructivistic idea that human cognitive processes are dependent on the knower's existing impressions and experiences of the world. This makes also the result of the process (the definitive understanding for the individual) subjective. As Polanyi explains, words and concepts themselves do not mean anything, for their meaning is in the personal understanding of every human being [19]. If we read a short message written on a paper, we probably remember the meaning of the message after a while but we might not remember the exact words that were written. We are generally not interested in the words themselves, but the meaning the words bear.

According to Polanyi's theory described so far, the nature of knowledge seems very subjectivist. However, Polanyi does not deny the existence of objective reality. Instead, he stresses that knowledge always has an objective side as well [19]. First, the linguistic character of thinking means that all human thinking comes into existence by mastering the use of language of certain society. According to Polanyi [20], this means that all thinking is rooted in society. Second, people are born in a certain culture. They grow up in some cultural environment and education typical to that community. This inevitably affects the way people see the world. Third, knowing something can be seen as a responsible act that seeks universal validity. Our knowledge often concerns the world around us, which means that we do not accept something to be knowledge unless we think that the knowledge corresponds to the state of matters in reality that is accessible to all. However, in Polanyi's thinking definite truth about reality is improbable because reality manifests an infinite amount of different possibilities to humans.

It may be possible that Polanyi overestimates the importance of language in human thinking. According to contemporary psychology, linguistic statements, and prepositional thinking in general, represent only one form of thinking. For example Paivio [16] has proposed that there is a nonverbal imagery system of thought alongside of linguistic processes. That does not, however, contradict with Polanyi's idea that human thinking is influenced by culture and tradition.

2.1 Two stages of awareness

The major feature of Polanyi's theory is a distinction between two stages of awareness in an act of knowing. *Focal awareness* concerns the conscious object of the directed attention. *Subsidiary awareness* provides the background within which the focal

awareness operates. An essential idea of the theory is that while attending to focal awareness a person *dwells in* subsidiary awareness that contains subsidiary components of the meaning of the focal target. Subsidiary knowledge is closely related to how the object of the focal awareness should be acted with.

Polanyi [19] describes the interaction between the two stages by an example of a pianist. While playing piano the pianist's focal awareness is attended to playing piano taken as a whole. The pianist knows subsidiarily, for example, how to move the fingers or how to read the notes while playing. The content of subsidiary awareness makes meaningful action with the target of focal awareness possible.

In an act of knowing the stages of awareness are interacting: the target of knowing is attended focally, which "activates" subsidiary knowledge related to the target. The target is then known based on tacit particulars, which enriches the meaning with personal understanding. Polanyi argues [19] that this kind of integration of focal and subsidiary awareness occurs in every act of knowing for it is necessary to the understanding of focal target. It is essential to recognize that the focal target gives rise to subsidiary knowledge in a sense that only the target may characterize subsidiary knowledge applied to it. Thus, the content of subsidiary awareness can only be tacitly applied by attending to the focal target. Practically this means that *subsidiary knowledge is not available for the knowing subject without a focal target*. This argument means that to be able to benefit from tacit knowledge a person has to have a context where tacit knowing is applied. The structure of tacit knowing is described in figure 1.

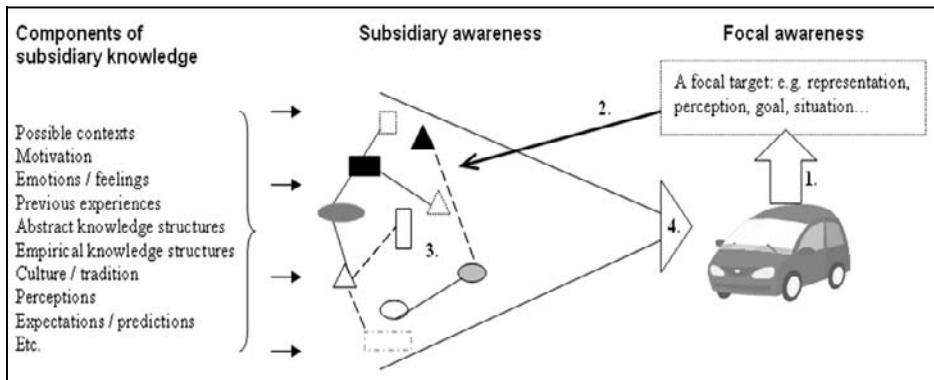


Figure 1. The process of tacit knowing: the target of knowing is attended focally (1), which enables the use of tacit particulars related to the target – the knower dwells in subsidiary awareness (2). Tacit particulars are integrated into a meaningful whole (3). A new meaning emerges as the target is attended from the subsidiary knowledge (4).

The two states of awareness are *mutually exclusive* [19]. This means that attention can be directed to only one of them at a time. If we try to direct our attention from focal awareness to subsidiary awareness, we interrupt the act of knowing; if the pianist shifts his attention from the playing to the movements of his fingers, he gets confused and he may have to stop. In addition, there should not be anything to attend to in subsidiary awareness anyway, because the meaning of subsidiary particulars is lost as a result of attending away from the focal target.

What then exactly is subsidiary knowledge? People gradually gain understanding of the reality as a result of personal experiences, conceptions and feelings. These kinds of components clearly constitute an essential part of subsidiary knowledge. Basically any kind of possessed knowledge may function as subsidiary, even knowledge that is focal in some other context. According to Polanyi [19], a mental effort has a heuristic effect in a sense that it tends to incorporate any available elements of the situation that are helpful for its purpose. This structure of knowing explains in a clear way the human ability to pack various features in an act of knowing and experience them simultaneously, which finally builds up the meaning.

We now present two examples (originally given by Polanyi) in order to illustrate how tacit knowing functions in practice according Polanyi's thinking.

Example 1: In a psychological experiment [11] subjects were presented tachistoscopically nonsense syllables. Half of the syllables had been associated with an electric shock. Subjects were unable to recognize the shock-causing syllables, but they showed symptoms of anticipating the shock (measured with galvanic skin response) at the sight of the shock-causing syllable. When subjects were asked to explain what made them expect the shock, they could not answer. Subjects learned to connect two terms, the shock and shock-causing syllables. They relied on their awareness of the shock syllables for attending to the shock, for the electric shock represented the meaning of the shock syllables. Thus, identification of shock syllables was impossible. The knowledge of them remained tacit. [21, pp. 7-8]

Example 2: All research must start from a problem that must be good and original. But how is it possible to see such a problem that no one else has seen before? If the problem is not seen before by anyone else, it seems to be either impossible to solve or meaningless, and cannot be good. However, as experience shows, scientific problems are found and solved. Thus, to be able to see a good problem, one has to see something hidden. One must have an inkling of a coherence of particulars of the problem. Indeed, the knower is guided by a deepening sense of coherence. This proves that we can know things that we cannot tell. [21, pp. 21-22]

2.2 Epistemological Significance of Tacit Knowing

The second example reveals the most striking argument of Polanyi's theory: if all the knowledge were totally explicit, there would not be a way to know problems or find their solutions. Thus, Polanyi denies the existence of totally explicit knowledge; all knowing is personal knowing requiring active and continued participation of the knower. He claims that all the spoken words, all the formulae and all the maps are strictly meaningless if they are deprived of their tacit coefficients [17, p. 195]. The denial of strictly explicit knowledge leads to the culmination of Polanyi's epistemology: all knowledge is either tacit or based on tacit knowledge. This means that *completely* articulated knowledge cannot exist.

Although in Polanyi's thinking knowing can never be completely objective, he recognizes the objective nature of things to be known, for there is an external reality and it is knowable to all. Consequently, Polanyi's view does not represent subjectivism or radical constructivism, but refers to realism. Polanyi's realism challenges the traditional definition of knowledge, according to which knowledge is seen as a

justified, true belief. The traditional view clearly emphasizes the objective ideal of knowledge.

In the next section we estimate to what extent Polanyi's ideas are present in the contemporary knowledge management literature concerning the externalization of tacit knowledge.

3. The Presence of the Original Meaning of Tacit Knowledge in Knowledge Management Theories

Nonaka and Takeuchi [14, pp. 11-12] describe a typical process of externalization with a following example (truncated):

In the late 70's top management of Honda (Japanese car manufacturer) realized that their car models were becoming too familiar. They started a project whose goal was to come up with a product concept fundamentally different from anything done before. The project team leader coined a slogan "automobile evolution" to express his sense of the project. The slogan posed a question: if automobile were an organism, how should it evolve? Based on this question, the team eventually developed an idea of a sphere: a car simultaneously short and tall. Such a car would be lighter, cheaper and more comfortable than traditional cars, they reasoned. The team called the product concept "tall-boy". The new concept provided the most room for the passenger and took up the least amount of space on the road. This process led to Honda City that was a revolutionary, urban car and a success product for Honda.

Nonaka and Takeuchi explain that this is a typical way that Japanese managers approach the process of making tacit knowledge explicit [14]. The use of figurative language and symbolism are important elements for the expression of inexpressible; the use of figurative language enables articulation of intuitions and insights, and further the distribution of tacit knowledge.

3.1 Relationship between Tacit and Explicit Knowledge

The idea of externalization of tacit knowledge is based on a classification of knowledge into tacit and explicit. This view seems problematic from the viewpoint of Polanyi's theory; Polanyi does not present different types of knowledge, but knowledge that always either has tacit elements or is wholly tacit. Knower constructs his understanding, and it does not happen automatically. Thus, knowledge has *both* explicit *and* tacit elements. In any case, explicit knowledge does not exist independently; tacit knowledge is an indispensable precondition that enables "explicit knowing".

As Nonaka and Takeuchi [14] recognize, tacit and explicit knowledge are not totally separate forms of knowledge. The classification is, however, present also in their theory up to a certain point, because it describes the conversion from one type to another. According to Polanyi's theory, tacit knowledge converted to explicit knowledge would still have a tacit side. Yet for example Kikoski and Kikoski [9, p. 65]

declare that “we need to recognize that there are two kinds of knowledge: the first kind is explicit and the second is tacit.” Similarly, for example Sivula’s *et al.* [22, p. 123] claim that “a common way of describing the characteristics of knowledge is to divide it into explicit and tacit knowledge as classified by Polanyi”, seems rather confusing. These kinds of arguments suggest that the concept of tacit knowledge has been quoted from Polanyi without knowing accurately the epistemological context it is brought from.

3.2 Unattainability of Subsidiary Knowledge

According to Polanyi’s thinking, the tacit dimension of knowledge is essentially related to subsidiary particulars and the way they are connected and then linked to the focal target. From this basis externalization of tacit knowledge does not seem possible by its definition: subsidiary particulars, of which the focally known part consists, cannot be known consciously as such because they only exist with the focus to which they are related. As Polanyi [19, p. xiii] states, “when we switch our attention to something of which we have hitherto been only subsidiarily aware, it loses its previous meaning.” This seems to mean that *tacit knowledge cannot be consciously known in itself*. Polanyi admits that an analysis may bring subsidiary knowledge into focus, but such specification would not be exhaustive. Polanyi [18, pp. 31] explains: “anything serving as a subsidiary ceases to do so when focal attention is directed on it. It turns into a different kind of thing, deprived of the meaning it had ... Thus subsidiaries are – in this important sense – essentially unspecifiable.”

3.3 Metaphors, Intuitions and Insights

An illustrative use of language is suggested an adequate method for externalization of tacit knowledge [e.g. 1, 14]. According to Nonaka and Takeuchi [14, p. 71], “appropriate metaphor or analogy helps team members to articulate hidden tacit knowledge that is otherwise hard to communicate.” But what kind of relationship exactly is there between a metaphor/analogy and the phenomenon it describes? It seems that a metaphor/analogy does not necessarily give an intensional definition, or even an intensional characterisation, of the phenomenon, but is predominately an example that merely represents the phenomenon in a certain way. Consequently, we might ask if externalization actually means giving *ostensive characterisation* based on that (tacit) knowledge. This would mean that no tacit knowledge is externalized but only some kind of a manifestation of it. According to Polanyi’s thinking, if we express a metaphor or give an ostensive definition, we still leave a gap to be bridged by the person who we are communicating to; we can only hope that the other person *himself* discovers the part we have not been able to communicate [21, pp. 5-6]. Polanyi seems to refer to the fact that a metaphor given by someone must still be decoded by the others. Consequently, tacit knowledge that is pursued to be passed from a person to another by the use of analogies/metaphors is unavoidably left behind in Polanyi’s thinking.

If we consider this idea by means of figure 1 presented earlier, we notice that an externalized concept appears and is placed in focal awareness. It is the result of the mental process that is enabled by and based on subsidiary knowledge. But why is the

result exactly that? What kind of processes and personal understanding led to it? It seems that these questions are almost impossible to answer despite the fact that the result is a reflection of knower's subsidiary knowledge. Similarly, an intuition may reflect tacit knowing of the knower, but the source of the intuition remains unarticulated. The knower cannot explain the processes an insight was based on.

However, metaphors and analogies certainly facilitate communication of things that are hard to describe as proposed; they indeed seem to provide an advantageous method for building a common ground for sufficiently similar understanding, discussion about experiences and knowledge creation. Instead of doubting that, we question the idea of *conversion* of tacit knowledge into explicit through such a procedure.

4. Conclusions

Tacit knowledge is often defined as knowledge that is *difficult, but not impossible to articulate*. Many authors, applying Polanyi's theory, claim that individuals' tacit knowledge should be externalized and shared in organizations. We argue that this view of tacit knowledge is too simplified. However, our point is not to criticize the theory organizational knowledge creation [14] that represents significantly different way of thinking and has already proven to be a useful tool in practice. Instead, we see it important to address varying use of the concept of tacit knowledge that often is claimed to be based on Polanyi's theory without actually being that so much. Based on Polanyi's thinking presented in this paper, we claim that tacit knowledge cannot be externalized in a way presented in the knowledge management literature. We claim that one important reason for the questionable use of the concept is that it has been separated from its theoretical background and then applied in a different kind of epistemological context.

Whereas Nonaka and Takeuchi never claimed that *all* tacit knowledge could be externalized, many authors seem to have interpreted them quite broadly. As Kikoski and Kikoski [9, p. 67] put it, "One of the major tasks of Information Era organizations that seek to be successful is to create the conditions whereby everyone can verbalize their tacit knowledge." In contrast, Polanyi seem to stress that human thinking, learning or generally any cognitive function cannot be resolved into logical or systematic collection of rules or assumptions - there probably is not any recognizable logic how tacit knowledge builds up.

Our understanding of the functions of human mind has increased enormously since Polanyi's days, thanks to the recent advancement in the area of neuroscience and cognitive science. We suggest that tacit knowing needs to be approached from those perspectives in order to understand its role in human behaviour better.

References

- [1] Ambrosini, V., Bowman, C., Tacit Knowledge: Some Suggestions for Operationalization. *Journal of Management Studies*, vol. 38 (2001), 811-829.
- [2] Argote, L., Ingram P., Knowledge Transfer: A Basis for Competitive Advantage for Firms. *Organizational Behaviour and Human Decision Processes*, vol. 82 (2000), 150-169.
- [3] Baumard, P., *Tacit Knowledge in Organizations*. Sage Publications, London (1999), 52-77.

- [4] Cook, S., Brown, J., Bridging Epistemologies: The Generative Dance Between Organizational Knowledge and Organizational Knowing. *Organization Science*, vol. 10 (1999), 381-400.
- [5] Descartes, R., *Discours de la méthode pour bien conduire sa raison, et chercher la vérité dans les sciences*. (1637).
- [6] Grant, R., The Resource-based Theory of Competitive Advantage: Implications for Strategy Formulation. *California Management Review*, vol. 33 (1991), 114-135.
- [7] Hori, K., Ohsuga, S., Articulation Problem – A Basic Problem for Information Modelling. Kangassalo, H., Ohsuga, S., Jaakkola, H. (eds.), *Information Modelling and Knowledge Bases*. IOS Press, Amsterdam (1990), 36-44.
- [8] Kangassalo, H., Are Global Understanding, Communication, and Information Management in Information Systems Possible? Chen, P., Akoka, J., Kangassalo, H., Thalheim, B. (eds.), *Conceptual Modelling - Current Issues and Future Directions*. Springer, Berlin (1999), 105-122.
- [9] Kikoski, C., Kikoski, D., *The Inquiring Organization – Tacit Knowledge, Conversation, and Knowledge Creation: Skills for 21st-Century Organizations*. Greenwood Publishing Group, Portsmouth (2004).
- [10] Klein, G., Moon, B., Hoffman, R., Making Sense of Sensemaking 1: Alternative Perspectives. *IEEE Intelligent Systems*, vol. 21 (2006), 70-73.
- [11] Lazarus, R., McCleary, R., On Subliminal Activation. *Psychological Review*, vol. 63 (1956), 293-301.
- [12] Locke, J., *An Essay Concerning Human Understanding*. (1689).
- [13] Mitchell, M., *Michael Polanyi*. ISI Books, Wilmington (2006).
- [14] Nonaka I., Takeuchi H., *The Knowledge-Creating Company – How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, New York (1995).
- [15] Orlikowski, W., Knowing in Practice: Enacting a Collective Capability in Distributed Organizing. *Organization Science*, vol. 13 (2002), 249-273.
- [16] Paivio, A. *Mental Representations: A Dual Coding Approach*. Oxford University Press, New York (1990).
- [17] Polanyi, M., *Knowing and Being*. Routledge & Kegan Paul, London (1969).
- [18] Polanyi, M., Logic and Psychology. *American Psychologist*, vol. 23 (1968), 27-43.
- [19] Polanyi, M., *Personal Knowledge: Towards a Post-Critical Philosophy*. University of Chicago Press, Chicago (1962).
- [20] Polanyi, M., *Study of Man*. University of Chicago Press, Chicago (1958).
- [21] Polanyi, M., *The Tacit Dimension*. Doubleday & Company, Garden City (1966).
- [22] Sivula, P., van den Bosch, F., Elfring, T., Competence Building by Incorporating Clients into the Development of a Business Service Firm's Knowledge Base. Sanchez, R., Heene, A. (eds.), *Strategic Learning and Knowledge Management*. John Wiley & Sons, Chichester (1997), 121-137.
- [23] Spender, J.-C., Competitive advantage from tacit knowledge? Moingeon, B., Edmondson, A. (eds.), *Organizational Learning and Competitive Advantage*. Sage Publications Ltd, London (1996), 56-73.
- [24] Tsoukas, H., Do we really understand tacit knowledge? Easterby-Smith, M., Lyles, M. (eds.) *Handbook of Organizational Learning and Knowledge*. Blackwell, Oxford (2003), 410-427.
- [25] Tuomi, I., Data is More than Knowledge: Implications of the Reversed Knowledge Hierarchy for Knowledge Management and Organizational Memory. *Journal of Management Information Systems*, vol. 16 (2000), 103-117.

Information System in Security Area Using Semantic Approach

Prof. Ladislav BUŘITA and Vojtěch ONDRYHAL

*University of Defence, Communication and Information Systems Department
Kounicova 65, 612 00 Brno, Czech Republic*

Abstract. The article presents results of an international project in the semantic approach to the information system development in security area. The aim of the project is the verification of semantic possibilities based on commercial software. The thesauri design and verification using document base and analytical SW is described, as well as UML applied to an ontology modelling. An ontology was implemented in ITM and IS was verified.

Keywords. IS, semantic approach, ITM, UML

1. Introduction

The international project on the information system (IS) in the state security area was started at the end of the year 2006. The project was prepared in cooperation with the Institute for Strategic Studies (ISS) at the University of Defence (UoD) and the Mondeca/France (www.mondeca.com) and Tovek/Czech Republic companies (www.tovek.cz), and it is a part of the research project [1].

The project is based on an application of the commercial software Intelligent Topic Manager (ITM), for the intelligent data organisation and retrieval. ITM is presented as an ontology-driven application and its meta-model can be expressed using OWL to formally define both its topic-map-like meta-classes, and specific data types. An ontology and knowledge base using ITM meta-model can be in both its RDF and XTM representations.

Simple meta-model

- Generic types: Topic, Association, Role, Data Item; defined by meta-classes.
- The meta-model is expressed in RDF-OWL, but also using Topic Maps concept.

Topics, Associations and Roles

- ITM semantic network uses a “hyper graph” structure; “nodes”: Topics (= vertices) and Associations (= edges), a node is either a Topic or an Association.

- Hyper graph “connectors”: Roles (= incidences). A Role links exactly one Topic to exactly one Association.
- Data Items can be attached to Topics, Associations and Roles.

Classes, meta-classes and workspaces

- Every ITM object has at least one declared class (or type). Associations, Roles and Data Items have exactly one type. Topics have at least one declared.
- Recursive meta-model: Classes and types are declared as topics, including meta-classes. Classes and instances are not defined in the same workspaces:
 - Basic Ontology workspace defines ITM built-in meta-classes.
 - Client Ontology workspace defines client classes.

The current state of the information processing in the ISS could be specified as decentralized and individual. The information obtained and created in the ISS is currently saved in the PC of an individual worker. The information is in the form of studies, articles, proceedings, presentations, academic documents and photos. They come from the Czech Republic and from international sources. The information subject classification is consistent with the subject of an individual group of ISS (security studies, warfare group, and resources – processes).

We suggested for the technical base an open software solution to achieve compatibility with SW ITM. The final state of the information processing in the ISS should be centralized and integrated. Save consolidated information in accordance to subject of ISS group, central management and integration, intelligent searching. The goal of the project is to develop a Prototype “Information System in the State Security”, to implement and verify it in the ISS environment. The prototype should allow conceptual searching, annotation creating, collaborating on knowledge, subject publishing according to selected criteria, exploitation of ontology and taxonomy. Project Phases:

- Installation of DBMS PostgreSQL, application of the server JBoss, SW ITM.
- Ontology research and preparation.
- Prototype building, implementation and verification.
- Results demonstration and evaluation.

2. Thesauri Design and Verification

The method of the thesauri design includes the preparation of the typical ISS document base and thematic vocabulary specification. Then the thesaurus was verified. The methods of thesauri verification:

- Analysis of the document base (text mining, harvesting), see Fig. 1.
- Thematic vocabulary corrections and thesauri definitions.

The analysis of thesauri terms was carried out using current information sources of the ISS that contain 12 145 documents. All suggested high level terms were retrieved upon the document based in SW Tovek Tools Analysts Pack. The first track was made by the 0,9 relevance of the term occurrence in the document, and the second track was made by the 0,75 relevance. Results are shown in the Fig. 1. The result of the analysis has discovered that one term (Art of War) is not relevant to the thesauri because there are a minimum number of occurrences. The next step dealt with the second level terms.

N	TERM	0,90	0,75	%0,90	%0,75
1	ARMAMENT	19	790	0,16	6,50
2	ARMY	185	3717	1,52	30,61
3	ART OF WAR	0	30	0,00	0,25
4	CAPABILITIES	42	3058	0,35	25,18
5	CONCEPTIONS	10	1570	0,08	12,93
6	CONFLICT	40	1810	0,33	14,90
7	DEFENCE	256	4928	2,11	40,58
8	DOCTRINE	46	689	0,38	5,67
9	ENVIRONMENT	19	1935	0,16	15,93
10	INTERERST	4	572	0,03	4,71
11	MODERNIZATION	33	1318	0,27	10,85
12	ORGANIZATION	31	2735	0,26	22,52
13	POLITICS	76	2540	0,63	20,91
14	POWER	5	513	0,04	4,22
15	RELATION	18	2114	0,15	17,41
16	RESOURCE	35	2602	0,29	21,42
17	RISK	85	1479	0,70	12,18
18	SECURITY	42	2677	0,35	22,04
19	SITUATION	24	3478	0,20	28,64
20	STRATEGY	47	1377	0,39	11,34
21	TERRORISM	7	464	0,06	3,82
22	THREAT	29	1185	0,24	9,76

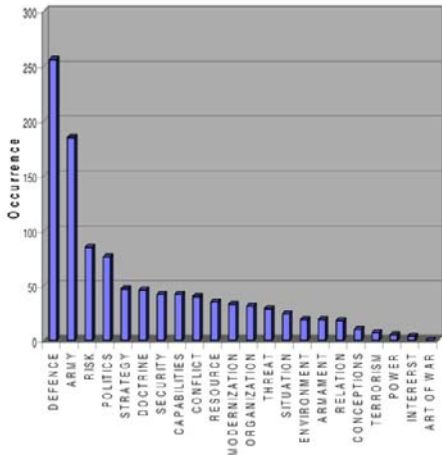


Figure 1. Thematic vocabulary.

3. UML (Ontologies vs. Database Approach)

UML (Unified Modelling Language) together with OCL (Object Constraint Language) and Structural (Data and Functional) Modelling provides other approaches to conceptual modelling of the selected domain [2]. The authors have been using these approaches for the information system with database modelling in the past 20 years.

In the next picture (Fig. 2), there is an example of the description of the project domain using a class diagram. Classes, attributes and associations from ontology have been transformed into similar elements in the class diagram. It is often stated [3] that UML approach does not have a first-class concept of an association; the association exists only between two classes and cannot be separated. For example, it is difficult to state that *owns* property in the RDF based ontologies “company owns vehicle” and “person owns dog” is the same.

As an alternative way to the association description the association class can be used. Association classes are not usually used very often, because they are not fully supported by CASE tools (e.g. code generation), but they can provide the missing complexity description of the association expected in ontology. The *owns* property for company and person can be similar because it can be derived from the same base class “OWNS”.

In our domain the following association classes have been added: AUTHOR-STUDY for the connection between AUTHOR and STUDY, AUTHOR_ARTICLE for AUTHOR and ARTICLE classes, PROCEEDINGS&JOURNAL-ARTICLES for PROCEEDINGS&JOURNAL and ARTICLE classes, KEY-TERM for KEY and TERM classes and finally TERM_ONTOLOGY-ARTILE for TERM_ONTOLOGY and ARTICLE classes.

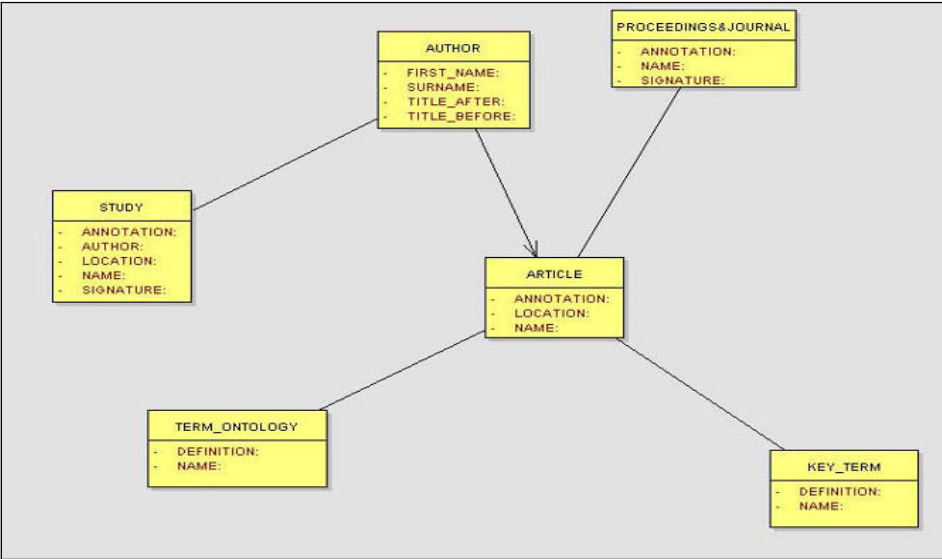


Figure 2. Ontology definition using UML.

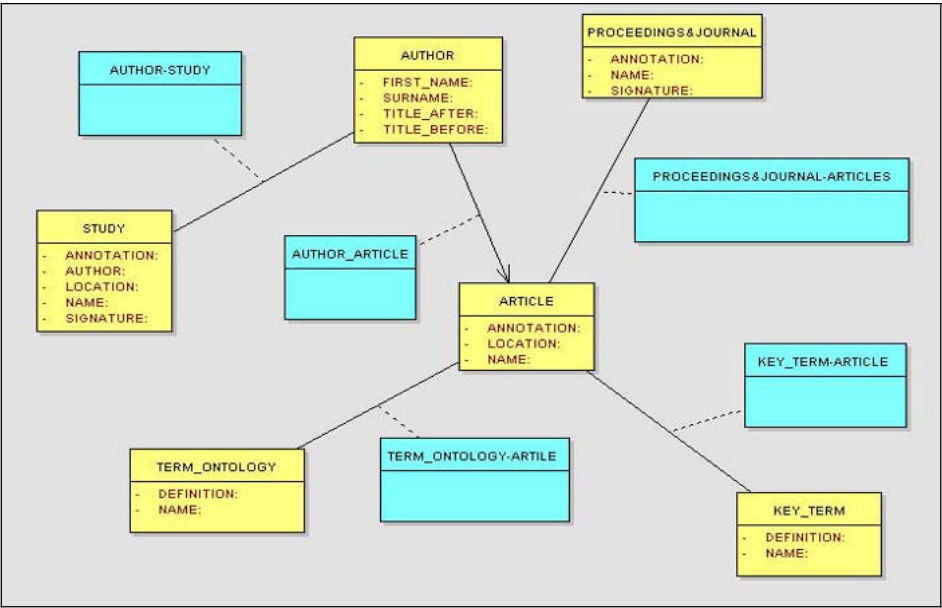


Figure 3. Ontology definition using UML with association classes.

4. Ontology Implementation and IS Development

Classes (CLASS), attributes (ATTRIBUTE) and associations (ASSOCIATION) together define the ontology in ITM. Similar components form the class diagram of conceptual data modelling.

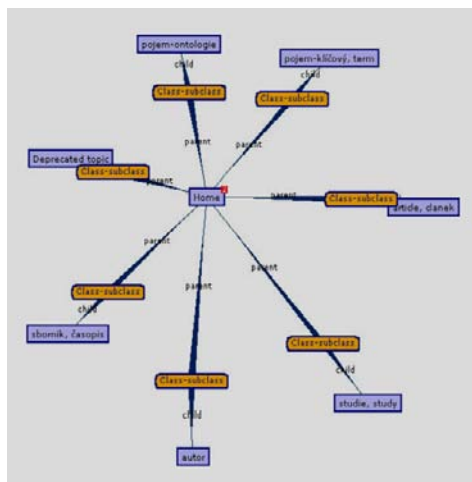


Figure 4. Ontology structure.

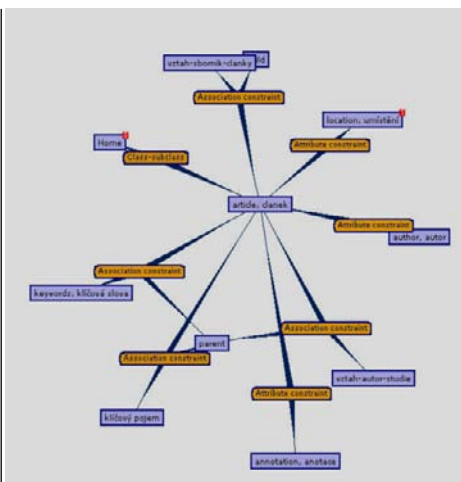


Figure 5. The details expanded for the class article.

The ontology for environment recording in the security information system defined in the project consists of the following classes: AUTHOR, ARTICLE, PROCEEDINGS, STUDY, KEY-TERM, and ONTOLOGY-TERM. KEY-TERM assigns article keywords into ontology, whereas ONTOLOGY-TERM associates thesaurus descriptors of the security field to the articles. On Figs 4 and 5 a part of the ontology structure is shown. The second figure represents details for a class that are hidden on high level diagram.

AUTHOR class

The AUTHOR class consists of author's name, surname and titles attributes. The AUTHOR-ARTICLE relation defines association with the ARTICLE class.

ARTICLE class

The ARTICLE class is responsible for creation, modification and publishing of all article records in the system. The ARTICLE class is constructed by NAME, ANNOTATION, AUTHOR and LOCATION attributes. The following associations are defined for the class:

- KEY_TERM-ARTICLE association provides connection for keywords within an article.
- TERM_ONTOLOGY-ARTICLE defines links with thesaurus.
- PROCEEDING-ARTICLES relation associates proceedings with articles.
- AUTHOR-STUDY connects authors with the article.

KEY-TERM class

The KEY-TERM class provides capability for creation, modification and publishing of all keywords defined within the STUDY and ARTICLE classes. The KEY-TERM class is defined by NAME and DEFINITION attributes. The KEY-TERM class can be associated with ARTICLE and STUDY classes.

TERM-ONTOLOGY class

The TERM-ONTOLOGY class includes thesaurus descriptors. These descriptors construct a conceptual searching tree. The instance of the TERM-ONTOLOGY class

should be assigned to an article or study, whereas instances of the KEY-TERM class may be unassigned. Needless to say, it all depends on the content of an article or study.

NAME and DEFINITION make the TERM-ONTOLOGY class. The TERM-ONTOLOGY class can be associated with the ARTICLE class.

PROCEEDINGS&JOURNAL class

PROCEEDINGS&JOURNAL class is a container of all articles published within PROCEEDINGS or JOURNAL and consists of the following attributes:

- NAME – name of PROCEEDINGS or JOURNAL,
- ANNOTATION – short annotation or description,
- SIGNATURE – unique marker.

PROCEEDINGS&JOURNAL class is associated with ARTICLE classes by PROCEEDINGS&JOURNAL-ARTICLE link.

STUDY class

STUDY class creates, modifies or publishes all records about the studies stored in the system, and is defined by the following attributes:

- NAME – name of the study,
- ANNOTATION – annotation or short description created for the study,
- AUTHOR – author of the study,
- LOCATION – place where study is stored and can be retrieved from,
- SIGNATURE – unique marker.

The instances of STUDY class can be associated with the instances of the KEY-WORD, KEY-TERM and AUTHOR classes (AUTHOR-STUDY). The implementation example is described on the figures below. The list of terms is on Fig. 6, the term definition on Fig. 7, and the hierarchy on Fig. 8.

Another example shows an article definition and context, including the link for the external location of the article.

5. Next Step – Application of the Oracle's Semantic Solution

Oracle's semantic web technologies constitute the first open, scalable, secure and reliable commercial data management platform for RDF and OWL-based applications [4]. Oracle Database 11g (Launched in September 2007) supports both RDF and OWL data management, allowing developers with the industrial leading software infrastructure for scalable and secure semantic applications. New object types based on a graph data model, RDF triples, are persistent, indexed, and queried, similar to other object-relational data types. These new enhancements ensure that application developers benefit from the scalability of Oracle Database to deploy scalable semantic-based enterprise applications.

The Oracle Database 11g semantic database features enable:

- Storage, Loading, and DML access to RDF/OWL data and ontology.
- Inference using OWL and RDFS semantics and also user-defined rules.
- Querying of RDF/OWL data and ontologies using SPARQL.
- Ontology-assisted querying of enterprise (relational) data.

Storage, Loading, and DML access to Semantic Data: Oracle Semantic Data Store allows storage, loading and DML access to RDF/OWL models. Each model is an

SUBJECT (108) : pojem-ontologie(108)			
			NAME
<input type="checkbox"/>			armáda, ozbrojené síly
<input type="checkbox"/>			bezpečnost
<input type="checkbox"/>			bezpečnostní doktrína
<input type="checkbox"/>			bezpečnostní konflikt
<input type="checkbox"/>			bezpečnostní politika
<input type="checkbox"/>			bezpečnostní riziko
<input type="checkbox"/>			bezpečnostní situace
<input type="checkbox"/>			bezpečnostní spolupráce
<input type="checkbox"/>			bezpečnostní strategie
<input type="checkbox"/>			bezpečnostní systém
<input type="checkbox"/>			cíle terorismu
<input type="checkbox"/>			doktrína
<input type="checkbox"/>			doktrína druhů vojsk
<input type="checkbox"/>			doktrína úřadů
<input type="checkbox"/>			doktrína sil
<input type="checkbox"/>			doktrína útočná
<input type="checkbox"/>			dvoustranný vztah
<input type="checkbox"/>			ekonomická moc
<input type="checkbox"/>			ekonomické riziko
<input type="checkbox"/>			ekonomický terorismus

Figure 6. Implementation example – list.

ARMÁDA, OZBROJENÉ SÍLY	
POJEM-ONTOLOGIE	
General information definition, definition Hlavní část ozbrojených sil České republiky. Jejím hlavním úkolem je zajištění vojenské obrany státu proti vnějšímu napadení a jinému útoků, které vyplývají z mezinárodních smluvních závazků České republiky a společné obraně proti napadení. Dále je předurčena k plnění úkolů v rámci mírových operací v oblastech nestability či konfliktů a realizaci záchranných a humanitárních akcí.	
Relations KLÍČOVÝ POJEM, VZTAHY-ONTOLOGIE	
KLÍČOVÝ POJEM child parent armáda, ozbrojené síly Členění operací s účastí sil Armády České republiky	
VZTAHY-ONTOLOGIE child parent armáda, ozbrojené síly ontologie	
VZTAHY-ONTOLOGIE child parent letectvo armáda, ozbrojené síly vojenská policie, četnictvo armáda, ozbrojené síly pozemní síly armáda, ozbrojené síly námořnictvo armáda, ozbrojené síly síly podpory a výcviku armáda, ozbrojené síly	

Figure 7. Implementation example – ontology-term definition.

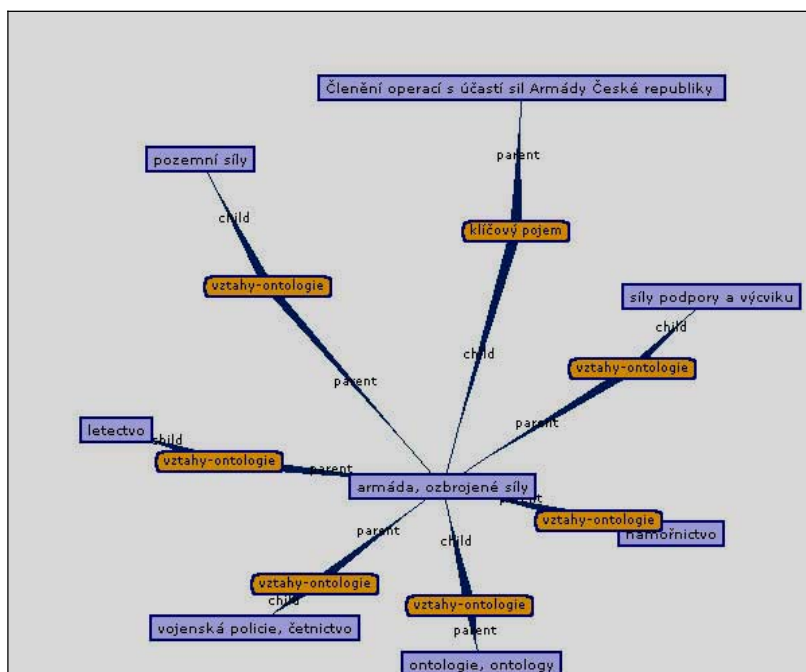


Figure 8. Implementation example – hierarchy.

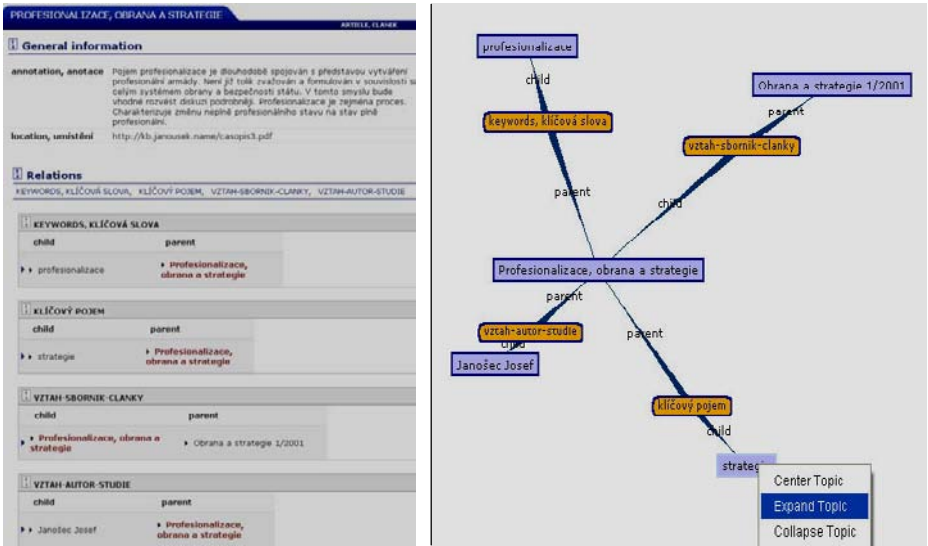


Figure 9. Article definition example and its relations – flat and tree views.

RDF/OWL graph consisting of directed labelled edges. The edge is labelled by a predicate and connects a subject node to an object node. A subject node must be a URI or a blank node, a predicate must be a URI, and an object node must be a URI, blank node, or literal.

Native Inferencing using OWL, RDFS, and user-defined rules: The ability to draw inferences from existing data using the precision and rigor of mathematical logic is probably the most important property that distinguishes semantic data from others. New enhancements include a native inference engine for efficient and scalable inferencing using major subsets of OWL. This OWL inferencing engine makes the existing native inferencing for RDF, RDFS, and user-defined rules more efficient and scalable. Inferencing may also be done using any combinations of these various entailment regimes.

Querying Semantic Data: RDF/OWL data can be queried using SQL. The SEM_MATCH table function which can be embedded in a SQL query has the ability to search for an arbitrary pattern against the RDF/OWL models, and optionally, data inferred using RDFS, OWL, and user-defined rules.

Ontology-assisted Querying of Enterprise (Relational) Data: Queries can extract more information out of relational data if the relational data is associated with ontologies in the domain of the relational data.

6. Conclusion

The user-friendly application for an effective access to ITM functions has not yet been implemented in our project. Unfortunately the project sponsor (ISS) changed their priorities and has stopped further development of the project according to the strategic changes in the internal information systems structure, thus the next part of the project is not required. In the future we are going to focus on the Oracle database environment

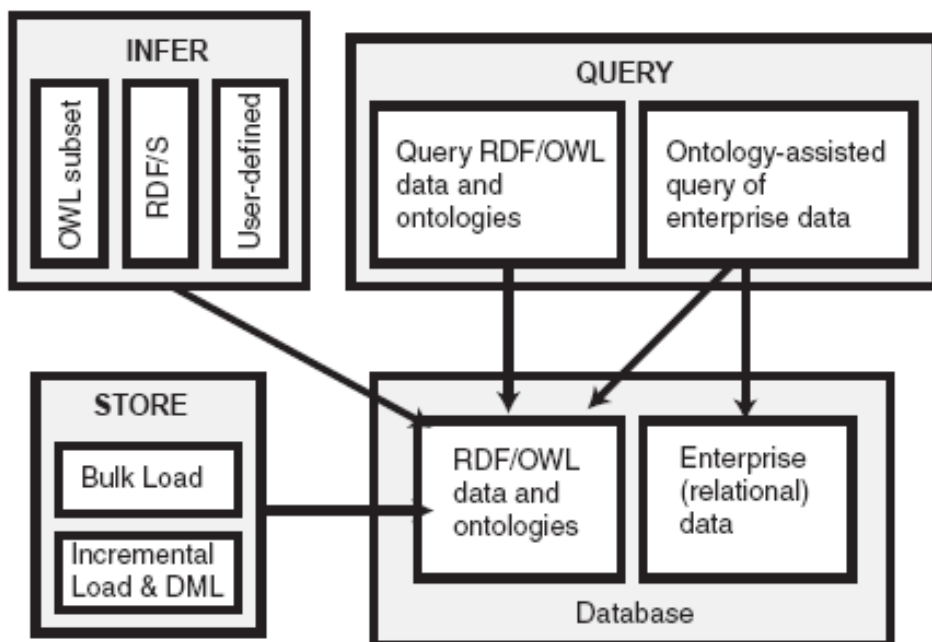


Figure 10. Oracle's semantic functionality.

and its new support to semantic web technologies, where the database and the semantic approaches could be compared more practically.

When we compare the development of information systems with the development using current technologies with relational database background and semantic technologies, we have to state that the semantic approach is more demanding. On the other hand, what is promising for the future is the information retrieval capabilities and robustness support for interoperability when using the semantic approach.

References

- [1] Development, integration, administration and security of CIS in NATO environment. Research project FVT0000403. Brno: UoD, 2004-2008.
- [2] Rumbaugh, J., Jacobson, I., Booch, G., The Unified Modelling Language Reference Manual, Second Edition, Pearson Education, Inc., Boston, 2005.
- [3] Kogut, P., Cranefield, S., Hart, L., Dutra, M., Baclawski, K., Smith J., UML for Ontology Development, White Paper, zdnet.com.
- [4] Oracle® Database. Semantic Technologies Developer's Guide. 11g Release 1 (11.1). B28397-02. Oracle, 2007, 96 pp.

How Psychology and Cognition Can Inform the Creation of Ontologies in Semantic Technologies

Paula C. ENGELBRECHT^{a,b} and Itiel E. DROR^b

^a *Ordnance Survey, Romsey Road, Southampton, SO16 4GU, UK*

^b *School of Psychology, University of Southampton, Southampton, SO17 1BJ, UK*

© Crown Copyright 2008

Abstract. This paper demonstrates how cognitive psychology can contribute to the development of ontologies for semantic technologies and the semantic web. Specifically, the way in which the human cognitive system structures and processes conceptual information can act as a model for structuring formal ontologies, and can guide knowledge elicitation and the use of controlled natural languages. One conclusion is that during knowledge elicitation the dynamic nature of human information retrieval needs to be taken into account to obtain an ontology that is appropriate for its context of use. A further practical implication is that ontology developers need to be more specific and explicit about what they mean by the term ontology (e.g. does an ontology describe typical concept attributes or attributes that are true of most instances?) when explaining the use of concepts in ontologies to domain experts.

Introduction

In computer science, ontologies are formal descriptions of sets of concepts related to specific domains [1]. They play a key role in semantic technologies and the semantic web; and are typically expressed using formal logic. The knowledge representation language for the Semantic Web, for example, is based on a family of logics (called description logics) that are decidable fragments of first-order logic [2]. In creating ontologies, ontology developers are faced with a variety of theoretical and practical problems. These include the representational formalisms for encoding ontologies, how to scope them and how they should be scaled and segmented [3].

This paper demonstrates how cognitive psychology can play a critical role in dealing with some of these issues. Specifically, it focuses on how our understanding of human knowledge representation and information processing could contribute to three aspects of ontology development. The first aspect to be discussed is the knowledge elicitation stage of ontology development during which domain experts act as knowledge sources. It is argued that knowledge engineers need to take into consideration both the strengths and the limitations of the human cognitive system in order to generate domain ontologies that are fit for purpose. The second aspect is concerned with the use of controlled natural languages to express ontologies. Evidence is presented about how the pragmatics of day-to-day language use (specifically, about how people use and understand generic and universal statements) can affect the content of ontologies au-

thored by domain experts. The third and final aspect illustrates how human knowledge representation can inform the structuring of ontologies.

1. Capturing Knowledge Representations (the Human as an Informant)

Domain ontologies are external representations of experts' subject-related knowledge. Knowledge engineers extract this domain knowledge using a variety of established knowledge elicitation techniques such as interviews, concept mapping and card sorting [4,5]. Due to its basis in formal logic, there is an inherent assumption within the discipline of ontology development that conceptual knowledge comprises relatively fixed and stable mental representations which can be accessed using the right knowledge elicitation techniques. This view is reflected in traditional approaches in cognitive psychology which assume that concepts are static mental representations that are retrieved from long-term memory when required [6]. However, there is evidence to suggest that internal representations of concepts are not stable entities but are the product of dynamic, context dependent processes [7]. This claim is supported by experimental findings which illustrate that peoples' conceptualisations of categories vary as a function of context and are unstable across time [7–9]. For example, it has been demonstrated that depending on the task context, category construction can be based on taxonomic (e.g. a sparrow is a bird) or thematic (e.g. cake and candles belong to the birthday party schema) relationships between concepts. In other words, depending on the context a "dog" can be classified as a mammal, a pet, a guide to the blind, a guard or a friend. This classification in turn will affect which "dog" attributes will come to mind. When thinking about dogs in the context of "kinds of mammals" the fact that dogs can serve as guides to the blind is less likely to come to mind than the fact that they bear life young.

The flexibility of internal representations of conceptual knowledge has several implications for knowledge elicitation. First, knowledge elicitation, rather than being a means of retrieving fixed concepts from memory, is an interactive process during which domain ontologies are modified and constructed (at least in part). This means that the construction of a domain ontology is of benefit to the domain expert, because engaging in this activity can solidify vague ideas and lead to novel insights. Second, the fact that internal knowledge representations are not fixed highlights the importance of setting the right context during knowledge elicitation. Concepts can encompass both context sensitive and context independent information [7]. Context-independent information about concepts (e.g. that robins are red-breasted) is considered to be highly accessible and relatively stable whereas context-dependent information is less accessible [7]. Knowledge elicitation exercises that are carried out in a neutral context are likely to elicit context-independent information. This might be desirable if one wishes to generate ontologies that can be used and reused across a wide variety of tasks and contexts. Frequently however, (especially in the case of semantic technologies) ontologies are intended to be used for specific tasks. On these occasions it is important to render the task context salient (for example by the way in which interview questions are worded) in order for the concepts and concept attributes that are most essential to the task to be elicited.

Third, in addition to contextual factors, other cognitive processes also affect knowledge representations. For example, a classical finding in cognitive psychology

shows that humans can only hold seven plus or minus two chunks of information in working memory at any given point in time [10]. Working memory capacity can be extended quite easily, however, by means of external aids. An implication of this for knowledge elicitation is that it is advisable to have written or visual representations of the discussed information available to both the knowledge engineer and the domain expert. The cognitive processes that come into play during knowledge elicitation can be very subtle. For example, it has been found that the process of comparing two similar categories (e.g. a donkey and a horse) leads to an increase in the perceived similarity between them even when differences are listed [11]. This finding would suggest that certain knowledge elicitation methods, such as card sorting (which involves the comparison of concepts in order to group them), may lead to the omission of defining attributes that are not shared with the comparison category. In order to capture domain ontologies that are fit for purpose knowledge engineers need to consider the above issues during knowledge elicitation.

2. Knowledge Representation (the Domain Expert as Knowledge Engineer)

Structured natural languages such as Sydney OWL Syntax [12], Attempto Controlled English [13] and Rabbit [14] are subsets of natural languages with restricted grammars and vocabularies. They enable domain experts to author ontologies without having to deal with the complexities of formal logic that underlie computational ontologies. Structured natural languages are relatively easy to learn because they harness people's pre-existing knowledge in language comprehension and production. An unintended consequence of this is that people are also likely to apply the same processes and assumptions that they use in day-to-day language use. A mismatch between how people understand structured natural languages and the assumptions inherent in formal logic can give rise to computational ontologies that contain erroneous statements; i.e. statements that do not comply with formal logic. This point is illustrated by considering the use of the universal quantifier in structured natural languages, and how "and" in English must sometimes be interpreted as "or" in formal logic.

Universal statements are generic assertions (e.g. "ducks lay eggs") to which a universal quantifier (e.g. "all", "every") is added. Structured natural languages use universal quantifiers to denote inheritance (e.g. "Every Man is a kind of Human") and other relationships that apply to all members of a category. Generic statements (e.g. "tigers are striped") express generalisations but lack quantifiers (e.g. all, some or most). A recent study on how people interpret generic and universal assertions [15] distinguishes three different types of statements: the first type consists of *characteristic* statements (that describe typical concept attributes), e.g. "ducks lay eggs", the second type refers to *striking* consequences, e.g. "ticks carry Lyme disease" and the third to statements that refer to the *majority* of instances, e.g. "cars have radios". The study described in [15] found that people tend to agree with generic statements that refer to characteristic properties (agreement on 89% of occasions) and, to a lesser extent, with generic statements that describe striking and majority properties (agreement on 68% of occasions). For universal statements, hardly any agreement was found for striking and majority properties (agreement on only 7% of occasions). However, participants had a tendency to agree with universal statements that describe characteristic properties,

e.g., “all lions have manes” (agreement on 47% of occasions), even though these statements were in fact false.

According to Khemlani and his colleagues [15], understanding generic characteristic statements is a cognitively primitive operation which is less demanding on the cognitive system than dealing with quantified assertions. The authors argue that people tend to process universal characteristic statements like generic characteristic statements. This argument, combined with the observation that counter examples to characteristic statements (e.g. “female lions do not have manes”) are not readily accessible, helps to explain the observed tendency to agree with false universal characteristic statements.

The study described above deals with language comprehension rather than production. Nevertheless, although the empirical question remains to be addressed, there is no reason to assume that the same processes would not apply to an ontology authoring process where people generate the characteristic examples themselves. Thus, the findings suggest that the use of the universal quantifier in structured natural languages, although it avoids a certain proportion of errors (namely those caused by the erroneous belief that striking and majority properties refer to all category instances), does not circumvent people’s tendency to agree with universal statements about characteristic properties. This tendency might cause domain experts to develop ontologies that contain erroneous concepts. That is, concepts which contain universal statements that do not apply to all instances of that concept (“all ducks lay eggs” for example).

There are ways of minimizing this potential source of “erroneous” concepts. For example, an ontology authoring tool which supports ontology authoring using structured natural languages could have inbuilt probe questions, e.g., “Can you think of a duck which does not lay eggs?”, to prompt the domain expert to access counter examples. Alternatively, the above findings could give rise to a reconsideration of the nature of ontologies. What can be said with certainty about all instances of a given category, and how meaningful and useful is this information? With the possible exception of inheritance, exceptions to almost every statement which initially seems true can be found. It might therefore be useful to consider how modal logic (which can express necessity and possibility) or default logic (which can express facts that are true in the majority of cases) could be combined with the description logics currently used in semantic web ontology languages like OWL. An early example of this is [16]. The utility of this would partly depend on the kind of use to which an ontology is put.

An awareness of how people deal with universal statements and other related issues should inform the development of languages and tools designed to develop domain ontologies. Another area of cognitive psychology that plays a central role in this context is research on errors and biases in human reasoning. For example, it has been found that people’s performance on syllogistic reasoning problems is affected by the believability of the conclusion [17]. This finding illustrates that humans find it hard to disregard their existing knowledge when performing abstract reasoning tasks. The manner in which domain experts represent and access conceptual knowledge and reason about it affects the ontologies they develop. It follows that what is known about natural language use and human reasoning should inform the development of both controlled natural languages and the ontology authoring tools in which they are implemented.

3. Logical Ontologies (the Domain Expert as a Model)

The most basic relationship in ontologies is the taxonomic (IS-A) relationship which denotes class inheritance, and has an analogue with property inheritance. Ontologies often contain taxonomic hierarchies of concepts [1]. Organising concepts into class hierarchies is both elegant and economic because relationships that apply to several related categories do not have to be repeated over and over again. For example, if one knows that all mammals breathe one does not need to encode this information again for cats; it can be inferred from the class hierarchy. Similarly, within the cognitive psychology literature it has been argued that conceptual knowledge is represented in the form of inheritance hierarchies in long-term memory [18,19]. According to this view the hierarchical structure represented in memory consists of IS-A links [20]. The model predicts that it should take people longer to infer that ‘a salmon is an animal’ than ‘a salmon is a fish’. This is because the verification of the former statement supposedly requires the traversal of several IS-A links, whereas the latter requires traversal of only one. Evidence in support of this hypothesis has been found for both category inclusion decisions and for the verification of property statements, e.g., a salmon can swim versus a salmon has gills [19].

An alternative interpretation of the above findings is that they reflect a computational process. This view assumes that because closely related concepts share many properties, the relationships between them are easier to compute [20]. The finding that it takes people longer to verify statements containing atypical than typical category members; e.g. it takes them longer to verify that a penguin is a type of bird than that a robin is a type of bird, lends support to this interpretation [20]. These and other findings have given rise to the conclusion that IS-A relationships arise from a set of computations rather than being a central component of semantic long-term memory [20,21]. The argument that human conceptual knowledge is not organised taxonomically is further strengthened by evidence (discussed in a previous section) which shows that concepts are context-sensitive representations that are assembled dynamically when needed [7,22].

The above discussion indicates that to model conceptual ontologies more closely to human conceptual representations requires shallow rather than deep hierarchies. The term shallow hierarchy refers to concept hierarchies with few taxonomic levels. Shallow hierarchies are less supportive of subclasses inheriting the properties of their superclasses than deep hierarchies simply because there are fewer taxonomic levels from which properties can be inherited. This means that the relevant attributes need to be associated to concepts directly rather than being inherited from concepts further up in the hierarchy. The use of shallow taxonomic hierarchies has several advantages in ontology construction. For example, although humans can happily categorise an instance as belonging to more than one category (e.g. a dog can be both an animal and a friend) best practice in ontology design does not encourage the explicit specification of multiple inheritance relationships; however, it is possible to give a category both animal-like and friend-like attributes. A further argument in favour of shallow hierarchies is that ontologies – as they are currently implemented – do not allow for non-monotonic inheritance. When making “dog” a subclass of “friend”, unwanted inherited properties of the “friend” category cannot be overridden. In sum, the psychological literature makes some suggestions about how to structure domain ontologies.

4. Discussion and Conclusions

This paper has highlighted two ways in which what is known about cognition and cognitive psychology can inform ontology development. First, several activities in ontology development (e.g. knowledge elicitation and ontology authoring) involve domain experts. The efficiency with which these activities are carried out and the utility of the resulting ontologies can be improved by considering human information processing and its limitations. Second, the human cognitive system in general, and human knowledge representation in particular, can act as a model for the structure of ontologies.

Cognitive psychology can make suggestions for improvements in how ontologies are developed, structured and used. For example, the above discussion illustrates that ontology developers need to be more explicit and specific about what a concept is. One possible solution would be to define a concept as something that holds true for all normal category members. This would avoid the interchangeable use of *characteristic* (typical) concepts and *majority* (true in most cases) concepts in ontologies. Doing so can help avoid non-trivial problems further down the line when ontologies are used for automatic reasoning; this is especially crucial when ontologies which make different assumptions are merged.

The above discussion has highlighted the need for a further investigation into the scope of ontologies. How does one decide which concepts and concept attributes to include in an ontology and which to leave out? Psychological theories of conceptual coherence can bear on this issue. It has been argued that conceptual coherence arises from peoples' theories about the world [23]. For example, people are much more likely to list "does not fly" as an attribute of "penguins" than "sharks" because peoples' knowledge of birds would predict that penguins (but not sharks) can fly. Thus peoples' theories about the world determine which attributes are important to a concept and which are not. As discussed above, peoples' concepts of a given category are unstable, context dependent, constructs. The fact that different theories about the world are salient at different times helps to explain this. An implication for ontology development is that the scope and purpose section of a domain ontology should constrain its theoretical framework. By doing so, the decision of which concept and concept attributes to include (and which to leave out) is made much easier.

A related problem to the above is how one decides at which level of granularity an ontology should be captured? Research in cognitive psychology has shown that humans favour basic level categories when thinking and talking about the world. This basic level of abstraction is a "compromise between the accuracy of classification at a maximally general level and the predictive power of a maximally specific level" [20]. In other words, basic level concepts are those which optimize both informativeness and distinctiveness [20]. Take for example the concept "Collie" (which is subordinate to the concept "dog"). Although "Collie" is more informative than "dog", it is also less distinctive (it is harder to tell a "Collie" from a "German Shepherd" than it is to tell a "dog" from a "cat"). The concept "animal" (which is a super-ordinate of "dog"), on the other hand, is highly distinctive (it is easy to tell animals apart from other concepts at the same level such as "plants" and "objects") but not very informative. Based on these observations the most appropriate recommendation might be to represent ontologies at the basic level and to move only one level up or down from it. Doing so will ensure that the hierarchy of the ontology is kept shallow. What constitutes the basic level of abstraction for a given ontology is very much dependent on the purpose for the ontology. For example, within the biology domain the concept "amino acid" would be basic

to an ontology describing proteins but not in an ontology that captures “cellular processes”.

Ontologies are intended for use both by computers and humans, yet they represent knowledge differently. For example, human conceptual representations are flexible and dynamic whereas logical ontologies are relatively fixed constructs. Furthermore, whereas human knowledge representations can deal well with uncertain and ambiguous class memberships logical ontologies (as they are currently implemented) cannot. Due to these and other inherent differences in the way in which formal logic represents knowledge and the way in which humans do, ontology development requires a careful balance between the needs of the human and the needs of the machine. In order to find optimal solutions to these problems, ontology developers need to be knowledgeable of the limitations of both humans and of formal logic.

References

- [1] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5, 199-220.
- [2] Smith, M. K., Welty, C. & McGuinness, D. L. (2004b). Owl web ontology language guide, W3C Recommendation 20040210. W3C. Latest version available at <http://www.w3.org/TR/owlguide/>.
- [3] Seidenberg, J. & Rector, A. (2006). Web ontology Segmentation: analysis, classification and use. In *Proceedings of the 15th International Conference on World Wide Web*, 18-22. New York: ACM Press.
- [4] Garcia Castro, A., Rocca-Serra, P., Stevens, R., Taylor, C., Nashar, K., Ragan, M. A. & Sansone, S. A. (2006). The use of concept maps during knowledge elicitation in ontology development processes – the nutrigenomics use case. *BMC Bioinformatics*, 7, 267.
- [5] Wang Y., Sure, Y., Stevens, R. & Rector, A. (2006). Knowledge elicitation plug-in for protégé: card sorting and laddering. In R. Mizoguchi, Z. Shi & F. Giunchiglia, *First Asian Semantic Web Conference (ASWC'06)*, 4185 of LNCS, pp. 552-565.
- [6] Rummelhart, D. E. & McClelland, J. D. (1986). *Explorations in the microstructure of cognition, volume 1: Foundations*. Cambridge MA: Bradford.
- [7] Barsalou, L. W. (1987). The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), *Concepts and Conceptual Development: Ecological and Intellectual Factors in Categorization*. New York: Cambridge University Press.
- [8] Roth, E. M. & Shoben, E. J. (1983). The effect of context on the structure of categories. *Cognitive Psychology*, 15, 346-378.
- [9] Lin, E. L. & Murphy, G. L. (2001). Thematic relations in adults' concepts. *Journal of Experimental Psychology: General*, 130, 3-28.
- [10] Miller, G. A. (1956). The magical number seven plus or minus two: some limits on our capacity for processing information. *The Psychological Review*, 63, 81-97.
- [11] Boroditsky, L. (2007). Comparison and the development of knowledge. *Cognition*, 102, 118-128.
- [12] Cregan, A., Schwitler, R. & Meyer, T. (2007). Sydney OWL Syntax – towards a Controlled Natural Language Syntax for OWL 1.1. In *Proceedings of OWLED 2007 Workshop on OWL: Experiences and Directions*, vol. 258 of CEUR Workshop Proceedings. <http://ceur-ws.org/>.
- [13] Kaljurand, K. & Fuchs, N. E. (2007). Verbalizing OWL in Attempto Controlled English. In *Proceedings of OWLED 2007 Workshop on OWL: Experiences and Directions*, vol. 258 of CEUR Workshop Proceedings. <http://ceur-ws.org/>.
- [14] Hart, G., Dolbear, C. & Goodwin, J. (2007). Lege Feliciter: Using structured English to represent a topographic hydrology ontology. In *Proceedings of OWLED 2007 Workshop on OWL: Experiences and Directions*, vol. 258 of CEUR Workshop Proceedings. <http://ceur-ws.org/>.
- [15] Khemlani, S., Glucksberg, S. & Rubio Fernandez, P. (2007). Do ducks lay eggs? How people interpret generic assertions. In D. S. McNamara & J. G. Trafton (Eds.), *Proceedings of the 29th Annual Cognitive Science Society*, 64-70. Austin, TX: Cognitive Science Society.
- [16] Klinov, P. & Parsia, B. (2008). Demonstrating Pronto: a Non-Monotonic Probabilistic OWL Reasoner. OWL Experiences and Directions DC 2008, Washington DC 1-2 April 2008. http://www.webont.org/owlled/2008dc/papers/owlled2008dc_paper_2.pdf.
- [17] Evans, J. St. B. T., Barston, J. L. & Pollard, P. (1983). On the conflict between logic and belief in syllogistic reasoning. *Memory and Cognition*, 11, 295-306.

- [18] Anderson, J. R. & Bower, G. H. (1973). *Human associative memory*. Washington DC: Winston.
- [19] Collins, A. M. & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Behavior and Verbal Learning*, 23, 625-642.
- [20] Murphy, G. L. (2002). *The big book of concepts*. Massachusetts: MIT Press.
- [21] Sloman, S. A. (1998). Categorical inference is not a tree: the myth of inheritance hierarchies. *Cognitive Psychology*, 35, 1-33.
- [22] Smolensky, P. (1991). Connectionism, constituency and the language of thought. In B. Loewer & G. Rey (Eds.), *Meaning in Mind: Fodor and his Critics*. Oxford: Basil Blackwell.
- [23] Murphy, G. L. & Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review*, 92, 289-316.

Knowledge, Accountability, and Relevance Systems – Objectivations of Social Reality through Shared Symbolic Devices

Hakim HACHOUR

Paragraphe Laboratory EA 349, Department of Information and Communication Sciences, University of Paris 8 – 2 rue de la Liberté, 93200 Saint Denis, France.

Abstract. This paper focuses on modeling collective knowledge objectivations. It is based upon a transdisciplinary approach and a comprehensive methodology. The proposed model is about the social settings of relevant meaning systems and the semio-pragmatic factors of knowledge objectivations. It is shown that the essential resource of practical action is routine knowledge which is objectivated in an arranged system of indications, products and signs: a symbolic device bound to the pragmatic motive of the ongoing socially organized activity and conditioned by a typified sociocognitive relevance system.

Keywords: communication, intersubjectivity, shared cognition, routine knowledge.

Introduction

The aim of this paper is to present an instructive model of knowledge objectivations processes. I have based my researches upon a transdisciplinary approach engaging with socio-phenomenology, analytical philosophy, ethnomethodology, systemics, and semio-pragmatics. In some manner, this diversity reflects the complexity of a natural phenomenon: the social creation and distribution of knowledge. Projects of “new sciences” highlight the modeling of human activities in order to understand, and then eventually compute, processes and practical actions [1]. The tertiarization of organizational activities had dynamized the researches on ‘immaterial goods’; as Drucker foresaw it, the primary resource of a purposeful organization, and the most productive one, is knowledge [2]. How knowledge is substantiated and socialized in everyday life? I suggest that this problematic could be understood through the study of symbolic mediations occurring during practices. Largely interested by collective design and management activities, I have been confronted with the problem of the objectivations of knowledge. This construction of ‘knowledge about knowledge’ has led me to pursue my researches on collective creativity and co-design processes through a point of view from where knowledge is produced-perceived-interpreted or ‘made accountable’: the in-group’s one. The following section of this paper consists of a brief synthesis of the epistemological and empirical foundations of the proposed model. In the second section, I will concentrate on describing more precisely the critical features of shared symbolic devices (SSDs) and the principal findings on collective knowledge objectivations. Finally, as a conclusion, I wish to discuss the question of ‘synthetic elements of knowledge’ in the light of the exposed model.

1. Epistemological and Empirical Backgrounds

1.1. Knowledge as a Sedimentation of Meaningful Experiences

The phenomenological interpretation of knowledge induces a relative definition: an element of knowledge is a pragmatically corroborated fact before being a true justified belief. Indeed, if, as Rescher exposed, someone believes something “on grounds sufficient to guarantee its truth and realizes this to be the case,” truth is “justifiedly believed”, and stems from social conventions [3]. This paper follows the definition of knowledge proposed by Schutz [4-7]. His model can be related to Nonaka’s synthesis of “tacit and explicit knowledge” (based on Polanyi’s theory) [8, 9]; these two models have two aspects very much in common: they both focus on the socialization of knowledge (through the perpetual succession of internalization and externalization processes) and they suggest that shared knowledge stems from shared practices [6, 8, 9]. However, the primordial distinction results from the modeling of an intermediary type that regroups “routine knowledge” [6]. This type refers to the practicability of everyday life with the aim to place emphasis on the essential importance of meanings-in-action. ‘Tacit knowledge’ (or the arrangement of purely subjective and experienced processes) is the part of knowledge which is totally taken for granted; this does not need, for the time being, to be analyzed further and forms the grounds of social reality: its “here and now”. On the other side, specific elements of knowledge are totally anonymized and thematized components of knowledge [6, 8].

Between these two idealized types, there is a less clearly defined one that is bound to practical purposes; it represents the interaction of gradual knowledge subtypes: *skills* (i.e., internalized routines of action ‘definitively’ established and totally bound to the corporality and the spatiotemporal conditions of the actor), *useful knowledge* (i.e., habitual evolved skills partially internalized and externalized) and *knowledge of recipe* (i.e., bodily detached routine that involves a more complex and explicit articulation of different skills and useful knowledge) [6]. This dialectical analysis of basic and specific elements of knowledge leads to the conceptualization of a balanced type grounded to the situation of knowledge acquisition, treatment and/or creation.

1.2. Observer’s Relevance System and Accountability of Knowledge

All practical action and socially organized activity aim at a goal, and the steps accomplished to reach this goal constitute a course-of-action. Action begins after making a decision, that is, the result of a choice among projects of action [4, 10]. This teleological character of all human actions involves several contextualization processes. Indeed, projects are made from an anticipation of next environments, and these are perceived and categorized via shared norms which consist in the application of ideal types of action. Contextualization processes proceed from the selection and treatment of relevant data to the situation and they are conditioned by the synergy of three relevance structures [6]: *thematic relevance* (a particular theme which can be imposed – by the unfamiliar, the social, the situation, the Others – and motivated by a voluntary change), *interpretational relevance* (the interpretive procedure of the theme which is imposed by the level of knowledge of the theme and motivated by the inadequacy between the theme and its knowledge), and *motivational relevance* (biographically conditioned motivations in the *because context* and teleological motivations’ chain in the *in-order-to context*). I do not develop these concepts here but it is needed to stress

some essential characteristics of the mutual interdependence of these three structures during practical actions. An element of knowledge proceeds from an interactive and synergetic process: during the acquisition of knowledge, subjective experiences – the configuration of the relevance system – are bound to a practical action and its result, that is, an act (see Figure 1); indeed, “knowledge is not a copy of the environment but a system of real interactions” [11].

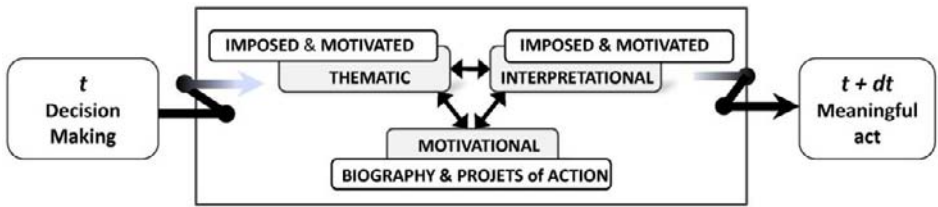


Figure 1. Relation between the structures of the relevance system and action (Schutz)

This setting of the relevance system is typified and institutionalized through time, feedbacks and repetition [6, 11] and is the ground of social action. During practical action, the adequacy of these mutually interdependent relevance structures for the ongoing course-of-action determines its routine or problematic definition. These exposed concepts bring out an essential question of accountable knowledge: in order to take into account a proposed element of knowledge, actors should be in a communicative environment and refer themselves to a common relevance system. During practical actions of everyday life, an account of knowledge could be apprehended as reflexive and indexical in Garfinkel’s sense: that is, “visibly-rational-and-reportable-for-all-practical-purpose” [12]. Indexicality could be understood in the terms of schemes attached to a socially organized activity, and reflexivity is a quality of interactional sequences which is used to conduct this activity by providing continual feedbacks whereby interactants actualize the meaning of their environment [12]. These two properties of accountable facts characterize their circumstantial and procedural factors. Thereby, the stock of knowledge (see Table 1) is produced by the temporal articulation and sedimentation of understood facts.

Table 1. Features of the elements of the stock of knowledge

Basic knowledge	Routine knowledge	Specific knowledge
Inner elements	Practicable network of elements	Outer elements
Attached to the situation	Contingent upon the situation	Detached from the situation
Subjective	Intersubjective	Objective
Tacit	Accountable	Explicit
Taken for granted	Circumstantial	Questionable

1.3. Methods and Fieldworks

Collected data are based on a comprehensive methodology [4, 6] engaging with complete participant observation, undirected interviews of the studied actors, retrospective analysis of conversations and interactions from audio/video recordings

and screen-captures (face-to-face and mediated communications), and in situ experiments; Garfinkel named the last of these “breachings” as 1) a relevant modification of the observer’s behavior that breaches the group members’ expectations; 2) the collection of altered behavioral feedbacks [12]. I focus my researches on the study of collective design activities with a special interest for these whose interactants are autonomous. The model exposed in this paper is based upon two fieldworks. The first of these was the study of musical collective design activities, with a special interest in musical communications because of their nonsemantic structures and their pragmatic involvement in design processes; I observed this group for a year and a half (more than 50 hours of interaction recordings). The second, which is still in progress, concerns collective projects design and management in a small company that provides consulting, training, and web services; I observed this group of four for eight months (10 hours of interaction recordings).

2. Collective Objectivations of Social Reality

2.1. A Pragmatic Typology of Objectivations: What Is a Shared Symbolic Device?

Schutz’s meaning and knowledge theories complement each other, indeed, according to him shared meanings are experienced, are elements of a social stock of knowledge, and are the presupposition for the detachment of the individual from the limitations of self-acquired knowledge [7]: a way to transcend oneself. Interaction analysis led me to distinguish three types of objectivations: indications, products, and signs. The common property of these types is that they are produced through appresentational relations (following Husserl’s development), that is, the association between a present datum of perception (e.g., information), and something at present not given (e.g., a meaning) [7].

Indications proceed from the experience of the other and define the interpretation of a datum as the result of a subjective process, whether it is true or not. Actors use *indications* to enhance their mastery of a situation by the reconstruction of hypothetical perspectives. *Products* are intentionally shaped and posited in a physical or virtual environment and combine three features with symptomatic degrees: *marks* (results of a purposeful modification of the environment), *tools* (objects used in order to change the environment) and *artworks* (aesthetically motivated transformations of the environment). *Signs* form complex systems resulting from the proactive socialization of an advanced products system (language is the most important); its use is the only way to objectivate specific elements of subjective knowledge which are detached to their spatiotemporal and social determinants. Thus, objectivations could be described as artifacts used for the communication of knowledge by reducing the mutual uncertainty of the interactants [7]. I use the adjective ‘symbolic’ to qualify this particular set of artifacts, which defines a ‘device’. Therefore, SSDs are shaped and used by social groups in order to normalize typical results of contextualization processes according to the ongoing activity (see Figure 2). The fact that routine knowledge (and its pragmatic motive) is made accountable through all types of objectivations shows the interest of their study. He who wants to formalize a knowledge system would have to collect and analyze objectivations that are recognized and ‘accountable’ from the in-group viewpoint. As archeologists do for example, it is possible to model convincing representations of knowledge and social systems by accessing only hypothetical *marks*, *tools* and *artworks* of a group: its *products*. The participation to the collective activity

permits to enhance this kind of representation: *indications* become available, so are the processes of specification of social reality into *signs* systems, and these systems themselves. The descriptive analysis of action's procedures completes that methodology by correlating the symbolic device and motivations.

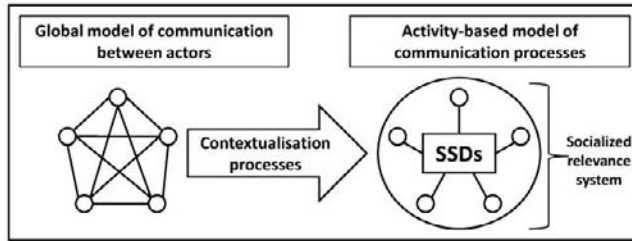


Figure 2. Global and activity-based models of communication processes.

2.2. Coextensive Meaning Domains and 'knowledge samples'

The periodic description of SSDs, which were made accountable during interaction, shows that they had evolved according to the social stock of knowledge. The objectivation of an element of knowledge could be apprehended as a sample which contextually symbolizes it, and accounts for it. It is now needed to discuss the question of the mode of symbolization. Each knowledge samples, e.g., 'readable data', could represent one or several denoted object(s), 'practicable meanings' to the other actors. Besides, Goodman has demonstrated that "The denotation is the core of representation and is independent of resemblance" [13]. Knowledge samples do more than simply denoting objects; they exemplify objects in a situation. The mode of symbolization called exemplification could be described through the notions of sample and label. A sample exemplifies only some coextensive label(s) of the object that it denotes, where exemplification is recursively "the reference of a sample to a feature of the sample" [13]. On one hand, the label of a knowledge sample is determined by the use of a situated SSD. On the other hand, the circumstances permit one to develop the SSD: these samples could coextensively exemplify multiple labels (such as its temporality, its practicality, or the experience of its subjective process of understanding).

This process of exemplification is understandable by the description of indexical expressions' meanings: the evolution of the denoted objects reflects changes in the SSD. The dynamic and reflexive adjunction of coextensive labels led the actors to collectively build an intersubjective meaning bound with the account of a knowledge sample. Whether the situation was defined as routine or problematic, the actors enriched their socialized relevance system with significant experiences, and thus, enhanced the mastery of their practices and evolved their cognitive representations. Consequently, routine knowledge seemed to lay a foundation for the generation of both basic and specific elements of knowledge.

2.3. On the Necessary Modeling of Communication Processes

The earlier developments bring me to expose some important conclusions. (1) As Wittgenstein realized, "language disguises thought" and its limits coincide with the limits of the paramount reality [14], with the boundaries of practical rationality [1, 7]. (2) The social stock of knowledge and the relevance systems are reflexively

constructed. (3) Collective objectivations of knowledge stem from communication processes, (4) and these later are conducted through the use of SSDs. Regarding these conclusions, I have to stress an important consequence: modeling organizational knowledge treatments must involve the modeling of communicational processes and therefore, the description of SSDs.

3. The Problem of Synthetic Knowledge

In conclusion, I wish to apply the exposed model to knowledge modeling practices. Human-readable knowledge bases currently designed are more precisely ‘knowledge’s samples bases’. The actor who deals with such knowledge bases has to label these samples in order to reconstruct the steps of the acquisition process and to simulate an adequate symbolic device. The problem with these synthetic elements of knowledge is that different courses-of-action – characterized by a typical “if..., then...” style [6] – can lead to the same objectivate problematic situation (‘now, I have *this* problem *and/or that* issue’), thus, the problem solving process, in order to be efficient, must take into account the polythetic sociocognitive and spatiotemporal articulation of the problem’s structure: its ontogenesis. Thereby, and in order to fit the practical purposes of the *knowledge seeker*, knowledge modeling practices have to make accountable the ‘historicity’ of knowledge samples.

References

- [1] Simon H. (1969). *The Sciences of the Artificial*. Cambridge: The MIT Press, 1996.
- [2] Drucker P. F. “The New Society of Organizations”. In HBR, September-October, 1992, pp.95 – 104.
- [3] Rescher N. *Epistemology – An Introduction to the Theory of Knowledge*. New York: SUNY, 2003.
- [4] Schutz A. (1932). *The Phenomenology of the Social World*. London: Heinemann Educational Books, 1972.
- [5] Schutz A. (1955). “Symbol, Reality and Society”. In *The Problem of Social Reality, Collected Papers I*, The Hague: Martinus Nijhoff, 1962, pp. 287 – 356.
- [6] Schutz A., Luckmann T. *The Structures of the Life-World, Volume I*. Evanston: Northwestern University Press, 1973.
- [7] Schutz A., Luckmann T. (1983). *The Structures of the Life-World, Volume II*. Evanston: Northwestern University Press, 1989.
- [8] Nonaka I. “Dynamic Theory of organizational Knowledge Creation”. In *Organization Science* Vol. 5, No. 1, February, The Institute of Management Sciences, 1994, pp. 14 – 37.
- [9] Nonaka, I., & Toyama, R. “Why Do Firms Differ? The Theory of the Knowledge Creating Firm”. In I. Nonaka & K. Ichijo (Eds.), *Knowledge Creation and Management – New Challenges for Managers*, New York: Oxford University Press, 2007, pp.13 – 31.
- [10] Teulier R., Lorino P. (dir.) and al. *Entre connaissance et organisation : l’activité collective – l’entreprise au défi de la connaissance*. Paris : Editions La Découverte, 2005.
- [11] Piaget J. *Biology and Knowledge: An essay on the relations between organic regulations and cognitive processes* (B. Walsh, Trans. From French). Chicago: University of Chicago Press, 1971.
- [12] Garfinkel, H. *Studies in Ethnomethodology*. NJ, Englewood Cliffs: Prentice-Hall, 1967.
- [13] Goodman N. *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis: Hackett, 1968.
- [14] Wittgenstein L. (1922). *Tractatus Logico-Philosophicus*. London: Routledge, 2001.

Inheritance and Polymorphism in Datalog: an experience in Model Management

Paolo ATZENI^a, Giorgio GIANFORME^{a,1}

^a *Università Roma Tre, Rome, Italy*

Abstract. We discuss the use of a Datalog extension that refers to a data model with inheritance in order to manage the generic dictionary of MIDST, our Model Management proposal for the generation of translation of schemas and databases from a model to another. In comparable scenarios, with structural similarities of predicates of the data model and syntactical and semantical similarities of rules, the use of hierarchies and a sort of polymorphism provide a significant simplification in the definition of complete translations (Datalog programs) and a higher level of reuse in the specification of elementary translations (Datalog rules) thus simplifying the development of such rule based systems.

Keywords. Model Management, Datalog, Polymorphism, Inheritance

Introduction

Datalog based languages have been recently used in various experimental projects in the database field [3,5,6,10,13,12]. In most cases, great benefit has been obtained, especially because of the simplicity of the language, of its declarativeness, and of the possibility of separating the “rules” that describe the problem of interest and its solution from the engine that implements the solution itself.

In this paper, we report on our experience in extending Datalog with features that handle inheritance and some form of polymorphism, and show that in this way it is possible to increase the effectiveness of the language and the degree of reuse of the individual rules.

The case study is represented by our model independent schema and data translation platform (MIDST), where data models and schemas are represented in a uniform way by means of a set of constructs (metamodel) and translations are coded in a variant of Datalog with OID invention. We introduce hierarchies in the metamodel, based on structural similarities of constructs, and extend Datalog in order to exploit such hierarchies: with our extension (based on directives for the rule engine and on the use of polymorphic variables) it is no more necessary to write a specific rule for each variant of constructs, but it is possible to write just one polymorphic rule for each root construct of a generalization; it will be the rule engine that will compile Datalog rules, substituting polymorphic variables and thus obtaining specific rules for each variants of constructs.

¹Corresponding Author; E-mail: giorgio.gianforme@gmail.com. Supported in part by Microsoft Research through the European PhD Scholarship Programme

The paper is organized as follows. In Section 1 we discuss related work. In Section 2 we illustrate our MIDST project and highlight the problem tackled in the paper. In Section 3 we introduce PolyDatalog, an extension of Datalog with concepts of polymorphism and inheritance and in Section 4 we draw our conclusions.

1. Related Work

The idea of extending logics and rule based systems with concepts like polymorphism, typing, and inheritance goes back to the beginning of 80's [14]. Recent approaches [1,2,8,9,11] adapt theories and methodologies of object-oriented programming and systems, proposing several techniques to deal with methods, typing, overriding and multiple inheritance.

Our approach differs from the aforementioned proposals. They introduce concepts of object-oriented programming and, in particular, propose overriding of methods for sub-classes, where needed; we have a different goal, we don't need overriding and don't define anything for sub-classes (sub-predicates, in our case). Instead, using object-oriented programming terminology, we define a method (the rule) for the super-class (the polymorphic construct) and, moving from it, generate specific methods (other rules) for the sub-classes (child constructs). From this point of view, our work has something in common with [7] where reusing and modification of rules is allowed by defining *ad-hoc* rules to substitute name of predicates involved in other rules.

2. The framework: MIDST

In this section we briefly introduce MIDST [3,4], our Model Management proposal for the generation of translation of schemas and databases from a model to another. We use MIDST in the following to introduce our extension of Datalog, but we want to remark here that such an extension, with the use of hierarchies and a sort of polymorphism, provides a significant simplification in the definition of complete translations (Datalog programs) and a higher level of reuse in the specification of elementary translations (Datalog rules) in comparable scenarios, with structural similarities of predicates of the data model and syntactical and semantical similarities of rules. More generally it can be used whenever the data of interest have characteristics that can be naturally modeled with hierarchies.

We use the notion of construct to represent and manage different models in a uniform way. Constructs with the same meaning in different models are defined in terms of the same generic construct; for example, 'entity' in an ER model and 'class' in an object-oriented model both correspond to the 'abstract' construct. We assume the availability of a universe of constructs. Each construct has a set of references (which relate its occurrences to other constructs, and may be optional) and boolean properties. Constructs also have names and possibly types (this is the case of lexical elements like attributes of entities in the ER model).

In MIDST, translations are specified in a Datalog variant, where the names of predicates are names of constructs and names of arguments may be OIDs, names, types, names of references and properties. A specific, important feature is OID-invention, obtained by means of Skolem functors.

After a large set of successful experiments, we realized the growing number of constructs and the increasing structural complexity of the metamodel generated a number problems with Datalog, mainly those mentioned in the Introduction: hard scalability and low reuse of rules.

3. PolyDatalog

In this section we illustrate PolyDatalog, an extension of Datalog with concepts of polymorphism and inheritance, which exploits structural similarities of predicates and classic rules.

Let us first of all observe that, in MIDST, in order to define a more expressive metamodel capable of properly representing a large number of models, we had to introduce new constructs, often just variants of pre-existing ones. Despite the fact that the number of constructs seems to be very contained, we remark that in the implementation of the metamodel in MIDST we collapse semantically identical constructs even if they are syntactically different because of various references; hence the global number of “real” constructs is high. Moreover, the needs to represent complex concepts, like structured elements or nested elements, triggered a higher structural complexity of the metamodel. The growing number of predicates (i.e. constructs in our scenario) and the increasing structural complexity of the metamodel cause the problems with Datalog mentioned in the Introduction: hard scalability and low reuse of rules.

Changing perspective and using terminology from software analysis and design, it is possible to consider every variant of a construct as a child of a generalization rooted in the construct that has only mandatory fields. The idea is that, in order to define rules for transformation of all variants of constructs it is not necessary to write a specific rule for each of them, but it is possible to write just one polymorphic rule for each root construct of a generalization; it will be the rule engine that will compile Datalog rules, substituting polymorphic variables and thus obtaining specific rules for each variants of constructs. In the general case, a polymorphic rule R for transformation of a construct C , when compiled, will be instantiated n times, producing n specific Datalog rules, R_1, R_2, \dots, R_n , one for each child C_1, C_2, \dots, C_n of the generalization rooted in C .

Reasoning on the structure of rules for specific constructs (i.e. for children of a generalization), we distinguish two kinds of similarities. Some rules have the same pattern modulo renaming constructs and Skolem functors and we refer to this case by saying that such rules are *syntactically analogous*. Other rules are isomorphic modulo changing Skolem functors and renaming constructs and Skolem functors; we say that such rules are *semantically analogous* to remark that, despite some syntactical difference, they have the same semantics. In these cases it is possible to write a polymorphic rule, with directives for the rule engine, which can compile such rule producing “standard” Datalog rules for each specific variant of construct, following a general procedure.

Let us introduce the main idea with an example from the MIDST metamodel mentioned in the previous section, and consider the simplified version of Object-Relational model illustrated in Figure 1 using our metaconstructs.

In this simplified Object-Relational model there are: *Abstract*, to represent typed tables; *Aggregation*, to represent tables; *StructOfAttributes*, to represent structured columns; *AbstractAttribute*, to represent reference columns from tables, typed or not, or

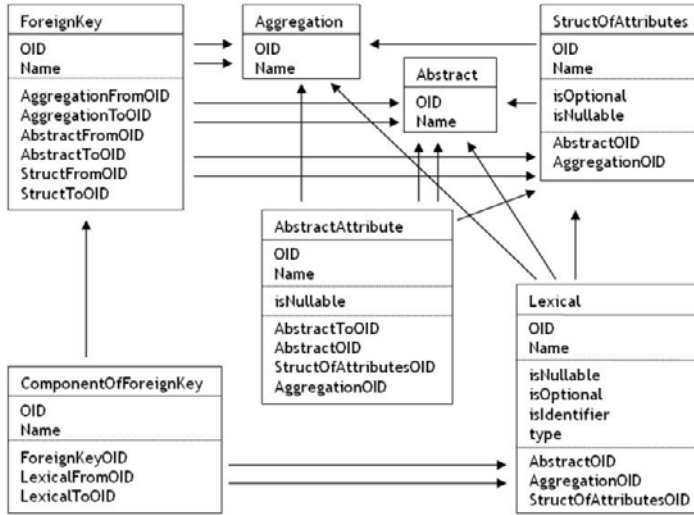


Figure 1. A simplified Object-Relational model

structured columns toward a typed table; *Lexical*, to represent columns of tables, typed or not, and structured columns; *ForeignKey*, to represent foreign keys from/to tables, typed or not, and structured columns; *ComponentOfForeignKey*, to represent columns involved in a foreign key.

Among these, *Lexical*, *AbstractAttribute*, *StructOfAttributes* and *ForeignKey* are constructs, with optional references, obtained collapsing semantically similar constructs: *Lexical* and *StructOfAttributes* have all references mutually exclusive; *AbstractAttribute* has the mandatory *AbstractToOID* reference and the others mutually exclusive; *ForeignKey* has two triples of references (those ending with “FromOID” and those ending with “ToOID”) whose elements are mutually exclusive. It is possible to introduce four generalizations, one for each of these constructs.

In the following we show the benefits of using inheritance in this scenario with the help of a case study: the transformation of a schema of the simplified Object-Relational model of Figure 1 into a Relational model with tables, columns, structured columns and foreign keys. In terms of our framework and hence in terms of constructs and rules, in order to perform such translation, we have to copy all elements not linked in any way with typed tables, like tables and their columns, structured columns of tables, foreign keys involving tables and their structured columns, and we have to properly transform typed table and elements linked to them, like structured columns of typed table, reference columns and foreign keys involving typed tables and their structured columns.

It is a pretty natural and correct guess that many of these rules are syntactically very similar to one another and semantically identical. Some examples follow. First, semantics of rules involving lexicals of “something” is always the same whichever is that “something”: transport the columns of various elements to the target schema, according to their belonging relation. Second, rules involving foreign keys have a unique goal: transport foreign keys to the target schema, according to the transformations undergone by elements linked by the keys themselves. Third, structured columns have to be trans-

formed according to the transformation undergone by objects they belong to: copied, if belonging to tables; transformed in structured columns of tables, if belonging to typed tables. The introduction of generalizations in the metamodel allow us to write just one polymorphic rule for each of these constructs, which can be compiled by the rule engine, producing “standard” Datalog rules for each specific variant of construct, according to the semantic aforementioned.

4. Conclusions

In this paper we have proposed an extension of Datalog, based on the use of hierarchies and a sort of polymorphism, that provides a significant simplification in the definition of complete translations (Datalog programs) and a higher level of reuse in the specification of elementary translations (Datalog rules) in scenarios with structural similarities of predicates of the data model and syntactical and semantical similarities of rules. We have integrated this extension in our MIDST project and the first successful experiments have supported our claim, since it enables higher level of reuse of Datalog rules and a substantial abatement of the number of Datalog rules constituting complete programs.

References

- [1] Serge Abiteboul, Georg Lausen, Heinz Uphoff, and Emmanuel Waller. Methods and rules. *SIGMOD Rec.*, 22(2):32–41, 1993.
- [2] F. Afrati, I. Karali, and T. Mitakos. Inheritance in object oriented datalog: A modular logic programming approach. Technical report, National Technical University of Athens, 1997.
- [3] Paolo Atzeni, Paolo Cappellari, and Philip A. Bernstein. Model-independent schema and data translation. In *EDBT*, pages 368–385, 2006.
- [4] Paolo Atzeni, Paolo Cappellari, and Giorgio Gianforme. MIDST: model independent schema and data translation. In Chee Yong Chan, Beng Chin Ooi, and Aoying Zhou, editors, *SIGMOD Conference*, pages 1134–1136. ACM, 2007.
- [5] Paolo Atzeni, Giorgio Gianforme, and Paolo Cappellari. Reasoning on data models in schema translation. In *FOIKS, LNCS 4932*, pages 158–177. Springer, 2008.
- [6] Philip A. Bernstein, Sergey Melnik, and Peter Mork. Interactive schema translation with instance-level mappings. In *VLDB*, pages 1283–1286, 2005.
- [7] Anthony J. Bonner and Tomasz Imielinski. Reusing and modifying rulebases by predicate substitution. *J. Comput. Syst. Sci.*, 54(1):136–166, 1997.
- [8] Gillian Dobbie and Rodney W. Topor. A model for sets and multiple inheritance in deductive object-oriented systems. In *Deductive and Object-Oriented Databases*, pages 473–488, 1993.
- [9] Gillian Dobbie and Rodney W. Topor. Representing inheritance and overriding in datalog. *Computers and Artificial Intelligence*, 13:133–158, 1994.
- [10] Georg Gottlob, Christoph Koch, Robert Baumgartner, Marcus Herzog, and Sergio Flesca. The lixto data extraction project - back and forth between theory and practice. In *PODS*, pages 1–12, 2004.
- [11] Hasan M. Jamil. Implementing abstract objects with inheritance in datalogneg. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 56–65, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
- [12] David Kensché, Christoph Quix, Mohamed Amine Chatti, and Matthias Jarke. Gerome: A generic role based metamodel for model management. *J. Data Semantics*, 8:82–117, 2007.
- [13] P. Mork, P.A. Bernstein, and S. Melnik. A schema translator that produces object-to-relational views. Technical Report MSR-TR-2007-36, Microsoft Research, 2007. <http://research.microsoft.com>.
- [14] Alan Mycroft and Richard A. O’Keefe. A polymorphic type system for prolog. *Artif. Intell.*, 23(3):295–307, 1984.

A Proposal for a User Oriented Language Based on the Lyee Theory

Keizo YAMADA, Jun SASAKI, Michiru TANAKA and Yutaka FUNYU

Faculty of Software and Information Science, Iwate Prefectural University, JAPAN

Abstract. In recent years, a problem has arisen in that users have become estranged from the processes of software development. End user development (EUD) is a solution to this problem. To realize EUD, however, we need to be able to describe the requirements of a user directly as a program.

In this paper, we propose a declarative language based on the Lyee theory for programming environments for users. Using our language, we describe a program based on subject-based programming in Lyee theory. The program is then executed iteratively.

Keywords. end user development, Lyee theory

1. Introduction

In recent years, a problem has arisen in that users have become estranged from the software development process. End user development (EUD), in which a user can describe software directly, is necessary to solve this problem. In Web application programs especially, the number of smaller scale systems that have developed in the short term, is ever increasing. Where a user develops the system directly, the development, from defining the system requirements to the program coding, is rapid. In this paper, we therefore propose a declarative language based on Lyee (governmental methodology for software providence)[5,6] theory as a user oriented language which supports EUD. We will design user interface to our language. Then, we propose a development model that the engineers can adjust the developed system after the users develop it.

Lyee theory is a software development methodology which includes a framework for describing a declarative program, which is divided into subjects. This is also known as subject based programming. The framework also includes the ability to calculate subject values by iteration, which involves a repeated execution order. A user can develop software directly by describing the requirements using the subject based language. However, no language based on the Lyee theory has yet been implemented, which is a big factor in preventing Lyee from becoming popular.

Several systems based on Lyee theory have been devised. LyeeAll 3 is aimed at larger systems. LyeeBuilder[2,3] is a development environment based on a formal system, namely the Lyee calculus[4], which uses a process algebra. It is implemented as a Java library and has a clear semantics based on process algebra. Hotaka et al. [1] provide an XML representation of software based on Lyee theory. However, it does not describe its details as a programming language. On the other hand, Several researches studied for

EUD using Lyee theory [8,9,12,13] such as a user requirement acquisition method[10] and a user input assisting system[11].

2. Previous works

Software development environments based on Lyee theory and developed in previous works are described below.

1. **Adjustable software**[9,13]: They propose a framework for user accessible boundary software such as an interface among a pure lyee program and a data base and a network.
2. **Requirement Acquisition Method**[10]: They give a system which analyzes the user's demand.
3. **Input assisting system**[11]: They give an input assisting system for LyeeAll 2 for developing information systems by an end user.
4. **Co-development system**[12]: They give an co-development environment among an engineer and a user based on the LyeeAll tool.
5. **LyeeAll 3**: Using this model, engineers can develop a large information system including many subjects of Lyee theory.
6. **LyeeBuilder**[2,3]: This is a development environment based on a formal system called Lyee calculus[4] which uses process algebra. It has a clear semantics based on mathematics and is implemented as a Java library in a multi-threaded environment.
7. **Hotaka et al.**[1]: They give an XML representation of software based on Lyee theory.

We propose a user oriented declarative language which reflects the characteristics of Lyee theory as a user oriented approach. We aim to develop a language, known as the *Lyee language*, which a PC-literate user can use.

3. Characteristics of Lyee theory and its development process

In this section, we describe the characteristics and development process of Lyee theory. In Lyee theory, the structure of software is very much simplified by representing the software as sets of the seven boxes depicted in Figure 1. The software development process using Lyee theory consists of the following 5 steps (as shown in Figure 2):

1. Define the requirements specification of the software.
2. Define the subjects which are extracted from the requirements specifications.
3. Define pallets and allocate each subject to an appropriate pallet.
4. Define scenario functions and assign a function to each pallet.
5. Define the process route diagram (PRD) for deciding the calculation order of the scenario functions.

Steps 3, 4 and 5 are necessary to decide the relationship between the subjects in Lyee theory. This makes the introduction of Lyee theory difficult, although it has superior characteristics.

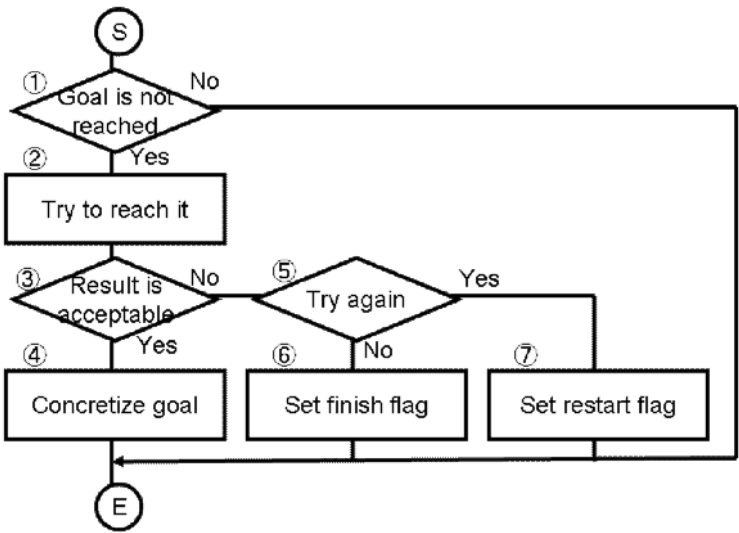


Figure 1. The seven box of Lyee theory

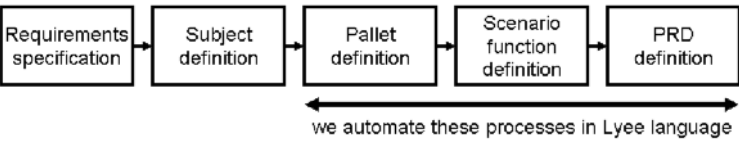


Figure 2. Software development process of Lyee theory.

4. Proposal of the Lyee language

In this section, we define the Lyee language, which utilizes the simple software structure of Lyee theory to shorten the development process of software. The purpose of our Lyee language is summarized in the following three points:

- 1. To utilize the simple software structure that Lyee theory proposes.
- 2. To shorten the development process.
- 3. To offer a user friendly method to a user who is not an expert in software development.

In our language, we design a program as a definition of subjects. Each subject belongs to a pallet. Each pallet is used either for output, for input or for calculating. We call these three pallets the scenario function. Our language controls the order of execution between scenario functions using a PRD.

Every subject has the seven box structure as in Figure 1, and is calculated in the following order:

- 1. In the second box, it checks whether the value of the subject has already been calculated or not. The subject does nothing if it is already calculated.
- 2. We give a definition of the subject in this box.

- 3. It checks whether the value of this subject is calculated normally or not.
- 4. If it is calculated normally, our language sets the result to the subject.

Each subject is allocated to a pallet. A scenario function consists of three pallets, namely a pallet W02 for input, a pallet W03 for calculations and a pallet W04 for output (Figure 3). The pallet W02 for input checks whether the input data from the screen is valid or not and casts the data type to an appropriate one. The output pallet W04 converts the data format to an appropriate one for displaying on the screen and selects the next screen. The calculation pallet W03 describes core processes of the software. The PRD controls the order of the scenario functions.

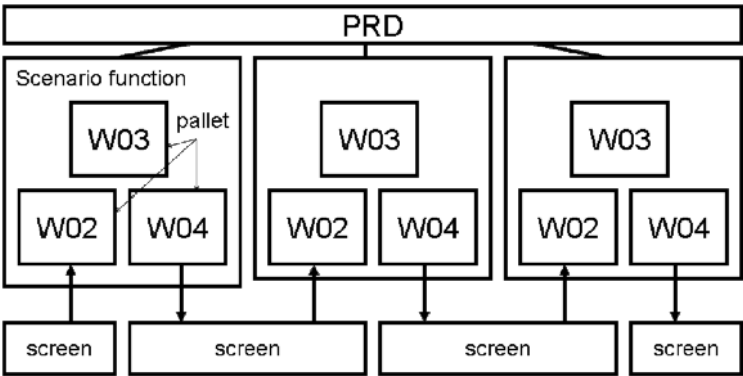


Figure 3. The outline of process route diagram (PRD)

In the language, the definition of a subject is represented by an equation. For example, the formula $a=b+c$ means that the definition of subject a is the result of the sum of subject b and subject c .

Our language can define the PRD automatically by describing the pallet nestedly. We can define the subjects in arbitrary order because our language controls the execution order of subjects by iteration. We can define the subject intuitively because we describe the subject imperatively.

Our language handles integer data, and possesses the ability to define a pallet. It can describe addition, subtraction, multiplication and division operations, logical formulae and conditional sentences. The syntax of the language is given in Figure 4.

A user defines subjects by equations from his requirements specifications and assigns them to an appropriate pallet. The user then obtains an executable program which satisfies the requirements specification, and the PRD of the Lyee program is generated automatically. Our system is simple and is suitable for small system development in contrast to LyeeAll3 and LyeeBuilder.

5. A working example and future plan

In this section, we give an example of the use of our language. A short program, prog, written in our language is shown in Figure 5. prog consists of six subjects a, b, c, d, e

```

program ::= pallet
pallet ::= 'pallet' name palletBody
palletBody ::= pallet | subject | '{' (pallet | subject)* '}'
subject ::= 'subject' name sentence
sentence ::= ifSentence | whileSentence | forSentence | foreachSentence
           | [Expr] ';' | '{' sentence* '}'
ifSentence ::= 'if' '(' Expr ')' sentence ['else' sentence]
whileSentence ::= 'while' '(' Expr ')' sentence
forSentence ::= 'for' '(' Expr ';' Expr ';' Expr ')' sentence
foreachSentence ::= 'foreach' '(' name 'as' name ['=>' name] ')' sentence
Expr ::= [Var '=' ] orExpr
orExpr ::= andExpr ('||' andExpr)*
andExpr ::= notExpr ('&&' notExpr)*
notExpr ::= '!' notExpr | CondExpr
CondExpr ::= AddExpr [( '=' | '!' | '<' | '>' | '<=' | '>=' ) AddExpr]
AddExpr ::= MulExpr (('+' | '-') MulExpr)*
MulExpr ::= Value (('*' | '/') Value)*
Value ::= '-' Value | '(' Expr ')' | Const | Var | FuncCall
Const ::= NumConst | CharConst | BoolConst | NULL
Var ::= name
FuncCall ::= name '(' ExprSeq ')'
ExprSeq ::= [Expr (',' Expr)*]
BoolConst ::= 'true' | 'false'
NumConst ::= Num+ [ '.' Num+ ]
Num ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
StrConst ::= '"' String '"'
NULL ::= 'null'

```

Figure 4. The syntax of the proposed language

and f. When this program starts executing, it iterates through the pallets to calculate the values of the subjects.

1. First, it executes pallet A and obtains $a=100$, and $b=50$. The value of subject d cannot be calculated yet because the value of subject c is unknown. It executes pallet A again iteratively, but obtains no new values. It then starts executing pallet B.
2. In the execution of pallet B, it obtains $c=1$. The value of subject e cannot be calculated yet because the value of subject d is not yet known. It executes pallet B again by iteration, but obtains no new values. Then it executes pallet C.
3. The value of subject f cannot be computed yet in pallet C, and thus it does not obtain any new values in this pallet. Execution therefore returns to pallet A.
4. In pallet A, it does not calculate the value of a or b, because they are already known. It obtains $d=200$ as a new value. With the iteration, it finds no other new values and thus moves on to execute pallet B.

```

pallet A {
  a = 100;
  b = 50;
  d = (1 + c) * a;
}
pallet B {
  c = 1;
  e = d * b;
}
pallet C {
  f = e;
}

```

Figure 5. A program written in the proposed language

5. It obtains $e=10000$ as a new value in pallet B.
6. It obtains $f=10000$ in pallet C.
7. Pallets A to C are executed once more, but there are no unsolved subjects, which can give new values and the algorithm ends.

6. Conclusion

In this paper, we have proposed a declarative language based on the subject based programming of Lyee theory. However, EUD is not completed only by this language. Also we need to extend our system including our language such as input assistant system.

Our future works include the following:

1. Developing our language to construct a Web application using subject based programming.
2. Extending our language to be able to access a database.

References

- [1] R. Hotaka, Y. Takeda and S. Toura: *XML Representation of Software Based on Information System Development Methodology Lyee*, IPSJ SIG Notes, **Vol. 2001-DBS-123-9**, No. 8, pp. 61–68, 2001 (in Japanese).
- [2] B. Ktari, M. Mejri and H. Fujita: *From Lyee-Calculus to Java Code*, In H. Fujita and V. Gruhn(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 283–300, IOS Press, 2004. Proc. of the 3rd SoMeT.
- [3] B. Ktari, M. Mejri, D. Godbout and H. Fujita: *LyeeBuilder*, In H. Fujita and M. Mejri(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 83–99, IOS Press, 2005. Proc. of the 4th SoMeT.
- [4] M. Mejri, B. Ktari and H. Fujita: *Lyee Methodology: A Formalization Using Process Algebra*, In H. Fujita and P. Johansson(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 235–261, IOS Press, 2003. Proc. of the 2nd Inter. Workshop on Lyee Methodology.
- [5] F. Negoro: *Principle of Lyee Software*, Proc. of the 2000 Inter. Conf. on Information Society in the 21st Century (IS2000), pp. 441–446, 2000.

- [6] F. Negoro and I. A. Hamid: *A Proposal for Intention Engineering*, Proc. of Inter. Conf. of Advances in Infrastructure for Electronic Business, Science and Education on the Internet (SSGRR2001), 2001.
- [7] H. Yagihashi: *Scientific Research into Establishment of Lyee's Word-based Program Structure*, Thesis of graduate school of Iwate Prefectural University, 2001 (in Japanese).
- [8] Y. Funyu, J. Sasaki, T. Nakano, T. Yamane and H. Suzuki: *An Experiment on Software Development of Hospital Information System*, In H. Fujita and P. Johannesson(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 328–340, IOS Press, 2002.
- [9] H. Suzuki, T. Yamane, T. Yoneda, J. Sasaki and Y. Funyu: *A Framework for User Accessible Boundary Software*, In H. Fujita and P. Johannesson(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 157–166, IOS Press, 2003. Proc. of the 2nd Inter. Workshop on Lyee Methodology.
- [10] T. Yamane, H. Suzuki, T. Yoneda, J. Sasaki and Y. Funyu: *A User Requirement Acquisition Method Based on Word-units for a User Development*, In H. Fujita and P. Johannesson(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 137–144, IOS Press, 2003. Proc. of the 2nd Inter. Workshop on Lyee Methodology.
- [11] K. Mitsui, K. Fujisawa, T. Yoneda, J. Sasaki and Y. Funyu: *HIMEKAMI: an End-user System development Environment Based on Lyee Theory*, In H. Fujita and V. Gruhn(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 361–371, IOS Press, 2004. Proc. of the 3th SoMeT.
- [12] T. Yoneda, K. Mitsui, J. Sasaki and Y. Funyu: *Co-developing Model for User Participation in Web Application Development*, In H. Fujita and M. Mejri(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 144–155, IOS Press, 2005. Proc. of the 4th SoMeT.
- [13] S. Gorlatch, T. Kameda, H. Fujita, M. Tanaka, Y. Funyu and O. Arai: *Towards Developing Adjustable Software: A Case Study with the Lyee Approach*, U In H. Fujita and M. Mejri(Eds.), *New Trends in Software Methodologies, Tools and Techniques (SoMeT)*, pp. 423–438, IOS Press, 2006. Proc. of the 5th SoMeT.

Towards Information Security Ontology in Business Networks

Jukka AALTONEN, Oliver KRONE, Pekka MUSTONEN

Department of Research Methodology, University of Lapland, Finland

Jukka.Aaltonen@ulapland.fi, Oliver.Krone@web.de, Pekka.Mustonen@ulapland.fi

Abstract. The existing business information and asset protection models are mainly based on organizational aspects of security management. Having the focal company as the unit of analysis makes it difficult to represent the complex network-wide phenomena, for example inter-organizational knowledge exchange and dynamic nature of security requirements. Here, a future research perspective is outlined, that makes ground to enable the design of ontologies applicable in the domain of knowledge exchange and information security in business networks. The topics that should be addressed include conceptual review of existing security models, analysis of business resource classification approaches, extending service oriented roles-linkage relationship models with network-wide security requirements, and the research perspectives in knowledge integration and requirements engineering.

Keywords. information security, business network, concept modeling, ontology

Introduction

In this research, a future research perspective is outlined that aims to provide a representation of the characteristics, requirements and features of a knowledge and information security (KIS) ontology in business networks. The security models and management approaches based on individual enterprises lack the treatment of the complex phenomena that emerge when the business is to be managed and conducted in a network context. The topics that are addressed include: (i) a conceptual review of existing information security models, (ii) network level asset identification by using novel business relationship models with adequate expressiveness to represent formally the inter-organizational transfer activities and (information system) service requirements to which they are based, and (iii) the research perspectives in organizational knowledge integration and requirement engineering.

1. Knowledge And Information Security

Along with the developments in information technology enabled business operations, management and decision making in an increasingly networked market environments, the knowledge and information dimension of business has become critical to organizations. The business management is confronted with decision making challenges that must take into consideration varying form of information from different sources. It is generally accepted that *information* is an asset that, like other important business resources, is essential to an organization's business and consequently needs to be suitably protected [1]. Especially in increasingly interconnected business

environments the issues in information exchange require special attention. It can be argued that in business context the management of *knowledge* emphasizes more the phenomena of understanding the meaning of the information (knowledge discovery and acquisition) and the issues of sharing knowledge intra- and inter-organizationally (knowledge exchange and knowledge integration).

In general, *security* is concerned with the protection of assets from threats, where threats are categorized as the potential for abuse of assets [2]. Threats can be caused by the environment, by human activities (malicious or accidental) or by vulnerabilities in the used tools, information systems, techniques or procedures. From the managerial perspective, then, it is especially the knowledge and the unique and critical capabilities and resources owned and utilized by the company (i.e. *information intensive assets*), that are to be protected to maintain the organizational autonomy and integrity. Thus, *information security* is the protection of information from a wide range of threats in order to ensure business continuity, minimize business risk, and maximize return on investments and business opportunities [1]. The method widely used in organizations to overcome the challenges of achieving security requirements identified during *risk analysis* initiatives is to use international and accepted security standards.

1.1. Security Standards

Information security standards are widely used by business enterprises to achieve the goals of risk analysis in form of organizational information security management policies and activities. The evaluation criteria for information technology security by the ISO/IEC 15408 standard [2] specifies also a general model of basic information security and assurance related concepts and relationships. According to ISO 15408 the information security is achieved by implementing a suitable set of controls, including policies, processes, procedures, organizational structures and software and hardware functions and that these controls need to be established, implemented, monitored, reviewed and improved, where necessary, to ensure that the specific security and business objectives of the organization are met [1]. The security assurance here means the techniques and methods by which the security requirements can be assessed.

A more recent security standardization effort is the ISO 17799 (lately renamed as ISO 27002) [1] that establishes guidelines and general principles for initiating, implementing, maintaining, and improving information security management in an organization. The standard contains best practices of *control objectives* and *controls* in the following areas of information security management: *security policy*, organization of information security, asset management, human resources security, physical and environmental security, communications and operations management, access control, information systems acquisition, development and maintenance, information security incident management, business continuity management compliance [1].

1.2. Generic Conceptualization of Organizational Information Security Models

A set of common terms and key concepts of an organizational knowledge and information security domain can thus be identified. These are:

- (i) ***business and assets*** - resources and other related entities that are of value to the business organizations (i.e. knowledge, information and physical resources)
- (ii) ***risk analysis*** - the act of identifying the potential dangers to critical organizational assets (e.g. assessment of threats, vulnerabilities, attacks, business impacts)

- (iii) **security controls** - the means of achieving security goals and requirements (confidentiality, availability, integrity, maintainability, usability) by using security standards and technologies, technical and organizational counter measures and protection approaches (security policies and best practices)
- (iv) **assessment** - ensuring that the security objectives are met by security evaluation, testing and auditing.

Following a multidisciplinary concept evolution (MCE) methodology and metamodel [3] to develop ontologies from simple terminologies and taxonomies, and using sources like above mentioned business security standards, published best practice documents, risk analysis approaches, information security glossaries and related concept models [1] [4] [2], the initial version of the generic organizational knowledge and information security (org_kis) conceptualization is presented (Figure 1).

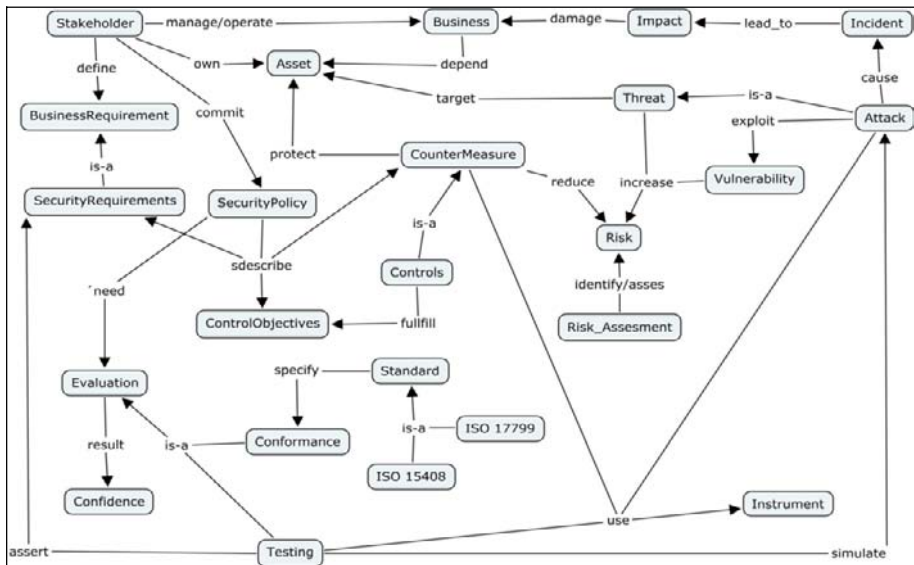


Figure 1: Generic organizational information security conceptualization

The conceptualization consists of three parts: (i) the key security concepts, (ii) the risk analysis and assessment related entities, and (iii) the security assurance perspectives like technical security testing approaches and practices [5] and standard compliance. The starting point for security management is the critical resource (Asset) identification, after which the threats and vulnerabilities that have business impacts above a tolerable risk acceptance level can be determined. Then, based on strategic information management and security policies the counter measures should be established. According to ISO 17799 [1], the counter measures (represented by the entities Control_Objective and Control) to threats are intended to be compliant with the requirements identified in a risk assessment (Risk_Assessment). In respect to organizational security management, the security policies and counter measures should be evaluated and tested by using a variety of security instruments (intrusion detection, packet sniffers, password crackers, etc.) simulating attacks (malware, breach, denial of service) that threaten to cause security incidents targeted against the organizational assets.

2. Information Security Ontologies In Business Networks

A security conceptualization developed above, encompasses most of the key elements of organizational information security but is lacking an adequate representation of these concepts in network context. Questions like, how are the critical network-wide resources (*network assets*) identified and who is the owner of them, or what is the relation between the individual intra-organizational security practices to the security requirements of dynamic knowledge and information exchange between the companies, need to be answered. Complexities to the information security phenomena arising in business networks are here addressed from the perspective of (i) service oriented roles-linkages in net-like structures, (ii) requirements engineering and (iii) MCE-based [3] ontology engineering.

2.1. Net-like Structures and Service Requirements

In contrast to the intra-organizational process based business view, where the resources are the input and output of business processes (*transformation activities*), the business network theories and economic market place studies consider the resource exchanges between companies (*transfer activities*) as one key element to structural analysis. In respect to the business network modeling, a service oriented extension of Actor-Resources-Activity model (ARA) [6] based roles-linkage model for business relationships has been proposed [7]. The approach conceives the business networks as a form of net-like structure and aligns the elements of these (environment, nodes, resources, dynamics and connectivity) to the information technology based service paradigm [8] elements (context, services, capabilities, interaction, composition). The business relationship analysis can then be based on a variation and an extension of the original roles-linkage model [9] where the linkages between companies are not only represented by their economic exchange functions but by a formal structure consisting of service requirements derived from inter-organizational interactions, information flows and process interfaces. In respect to the security needs of these knowledge and information exchanges, it is here proposed that the dependency of transfer activities (and the related assets) on the trust and power relationships of business partners and roles in the network could be analyzed and be included as part of the formal service requirement linkage structure (for example, in form of information security control objectives or end-to-end inter-organizational security policy representations).

2.2. Requirements Engineering in Networks

In a prior study that concentrated on a conceptional level on the development and implementation of information systems (IS) in net-like business settings, a layered approach to the obstacles of implementation was examined [10]. There it was suggested to treat and analyze requirements for those information systems on the level of the individual organization that is a member of the network, organizational level analysis being in line with traditional requirements engineering (RE), and on the overall level of the network. Reason is that only at the level of the network individual organizations' contributions to the production of a shared good or service, which is happening in a mutual process, can be identified.

It is suggested that the IS requirements in networked business settings take their origin in the individual organizations' and their processes. These organizational processes are then consolidated by a combination of the process outputs that the different service providers render to the network. Therefore it seems as if in the process

of RE first organizations have to obtain an understanding of their internal processes, and their outputs. For this reason there is a strong urge to align process components in the service provision among the contributors, which in essence means that also on the overall service provision RE has to be performed [10].

The process output exchange activities among the actors in the network represent the *transfer activity*, generating interwoven process flows. For a network-level knowledge and information security ontology this process based view is important, because it allows to asking how to identify an owner of the process. Reason is that the network output (*network asset*) is transparent in respect to the ownership, as it is result of a joint activity.

Thus there is need for methodologies that could identify the main owner of this output. This owner can also define the information security requirements in respect to this shared output; “ownership” of a *network asset* and information security requirements become interdependent objects. In this perspective not necessarily the largest process input in kind justifies definition of overarching network security requirements, but rather the impact of the input to the overall network output.

2.3. Ontology Engineering

The development of knowledge and information security ontology for business networks (*net_kis*) is based on the one hand on the general ontology engineering methodologies and on the other hand on the above representation of generic conceptualization of existing information security models and the discussion about the possible solutions to address the emerging complexities of the *network effect* to the entities and relations of intra-organizational concept models.

The *ontology engineering* is an emerging discipline of semantic web based technologies to design and examine theories, models and methodologies in order to provide support to the construction of formal knowledge representation frameworks, environments and repositories based *ontologies* (i.e. an explicit specifications of domain knowledge conceptualizations [11]). In the domain of information security the benefits of developing and utilizing ontologies include:

- the specification of semantic relationships between different organizations security conceptualizations
- possibility to extract and link domain dependent entities to existing informal and semi-formal security concept models
- organizational security awareness and information and knowledge sharing between stakeholders by aiding in mutual understanding of main terms and concepts and in agreeing about the most important security aspects
- enabling the management and maintenance of security requirements and risk analysis
- possibility to construct a business network wide shared security knowledge base in form of KIS ontology repository

The ontology engineering specific *competency questions* could here be explicated by identifying the business and IT usage scenarios of security ontologies in more detail, and by interviewing the stakeholders of business networks and non-commercial community users (developers, software engineers, domain experts and business users, conformance evaluators, auditors and testers).

3. Discussion And Conclusion

In line with the general idea that the context of a concept model has a fundamental effect on the concept analysis and mapping functions of the entities and relationships of the conceptualizations, it is here proposed that the universe of discourse of a security ontology is determined and influenced by the business network domain area analysis.

3.1. Unit of Analysis and Security Conceptualizations

In industrial network research terminology, the unit of analysis in the generalized organizational conceptualization of existing security models (*org_kis*) is the focal company (i.e. a business enterprise). Correspondingly, in case of the network wide security ontologies (*net_kis*, to be developed based on this research) the unit of analysis is the network. Considering the conceptual neighborhood of the entity representing the organizational assets (*org_kis:Asset*), and especially the relation that connects the owner to it (Stakeholder own Asset), it is evident based on the above discussion and the suggested *network effect* on the organizational information security perspectives, that changing the domain of discourse from focal company to network, the analysis of the concept of *network asset* is not identical to the conceptualization having a broader context (*net_kis*). Also the relation to the asset owner has to be reconsidered in the network context, e.g. along the lines outlined in the requirements engineering section above. It is suggested that the inter-organizational transfer activities are the main cause of this phenomena (see Figure 2). In networks the interactions of the participating organizations are represented by linkages (Linkage connect Organization) that depend on transfer activities (Transfer_Activity) during which the organizational assets (*org_kis:Asset*) are exchanged and thus the network level entities are generated (e.g. *net_kis:Network_Asset*). Such emerging entities have a different semantics in respect to original conceptualization (e.g., how is the owner to be determined in the network level).

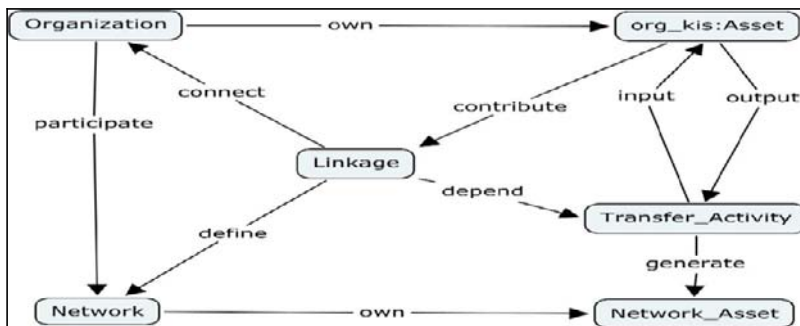


Figure 2: Network effect: transfer activities generate network-level entities

3.2. Future research

Above, the issues in existing intra-organizational security conceptualizations were outlined and the conceptual effects of network level business interactions to especially to asset identification and related security considerations were discussed from service oriented roles-linkage model and requirements engineering perspectives. However, the intention of this work was to unveil and explicate the open research issues to be

conducted as part of the future knowledge exchange and information security related research projects. Thus, a more elaborate examination of the main characteristics of inter-organizational knowledge and information security (net_kis) ontology should focus on the following topics:

- intra-organizational knowledge set specification
- specification of information flows generated in the transfer activities between organizations including also the meta-data of tangible assets (material flows composed of physical resources)
- definition and implication of knowledge transfer to collaborating agents
- impact of legal contracts and between stakeholders to KIS requirements
- dynamic nature of security considerations (security classification can change depending on the phase of the business transaction)

3.3. Conclusion

The objective of this research was to present the requirements and features for constructing a generic service oriented knowledge and information security ontology applicable in inter-organizational business interaction environments. In summary, the plausible way to conduct the analysis of the suggested network effect and to extended it to all of the entities present in the organizational security conceptualizations is to use such business network modeling approaches that enable the collection of information in form of formal service extended roles-linkage matrix representations. This should be done in order to reveal other possible dependency patterns between networked business relationship linkage types and the security conceptualizations and considerations of the related knowledge and information exchanges.

References

- [1] ISO/IEC JTC1/SC 27 (2005). ISO 17799 - Information technology - Code of practice for information security management. International Organization for Standardization (ISO).
- [2] ISO/IEC JTC1/SC 27 (2005). ISO 15408 - Information technology - Security techniques - Evaluation criteria for IT security. Part 1: Introduction and general model. International Organization for Standardization (ISO).
- [3] Aaltonen, J., Tuikkala, I. & Saloheimo, M. (2007). Concept Modeling in Multidisciplinary Research Environment. In H. Jaakkola, Y. Kiyoki & T. Tokuda (Eds.), *Proceedings of the 17th European-Japanese Conference on Information Modeling and Knowledge Bases (EJC 2007)* (pp. 143-160) Pori Tampere University of Technology, Pori.
- [4] C & A Security Risk Analysis Group (2003). *Qualitative Risk Analysis Model*. <<http://www.security-risk-analysis.com/introduction.htm>> Last accessed: 16.01.2008
- [5] Mustonen, P. (2007). *Implementaion Model of a Secure and Efficient ASP-system*. University of Oulu, Oulu, Finland.
- [6] Håkansson, H. & Snehota, I. (1995). *Developing Relationships in Business Network*. : Routledge.
- [7] Aaltonen, J. (2007). Service oriented extension of roles-linkage model for travel business networks. In M. Saloheimo (Ed.), *Operative Network Integration: working papers from research project, University of Lapland, Department of Research Methodology Reports, Essays and Working Papers* (pp. 166-181) Rovaniemi, Finland University of Lapland Press.
- [8] OASIS (2006). *Reference Model for Service Oriented Architecture*. (SOA Reference Model Technical Committee). 01.08.2006: OASIS
- [9] Kambil, A. & Short, J.E. (1994). Electronic Integration and Business Network Redesign: A Roles-Linkage Perspective. *Journal of Management Information Systems*, 10,59-83.
- [10] Krone, O. (2007). Information System development in a Business "Network". In M. Saloheimo (Ed.), *Operative Network Integration: working papers from research project, University of Lapland, Department of Research Methodology Reports, Essays and Working Papers* (pp. 122 -138) Rovaniemi, Finland University of Lapland Press.
- [11] Gruber, T. R. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, 43,907-928.

The Mouth Articulatory Modelling and Phonosemantic Conceptualization as In-Formation of Human Language

Alexei MEDVEDEV
University of South Australia
meday001@students.unisa.edu.au

Abstract. The work claims the trilateral unity (sameness) of sound, thought, and mouth gesture in human speech. This fundamental principle allows modelling of the world's objects by analogy in their – mostly spatial – traits and the mouth gestures for the articulation of sounds. The *phonosemes* – a newly introduced term for the sounds of human speech as bearers of intrinsic meaning – appear to be universal across all human languages and comprise the multilingual diversity of word formation as various instances of the primordial conceptualization of knowledge as the *in*-formation, a sort of Mentalese.

Keywords. Universality, mouth articulation, sounds meaning, spatial analogy, *phonosemes*, Mentalese

Introduction

Human language expresses human concepts by means of words of sounds. The work attempts to explain the relationship between sound and meaning (thought) in human language as the multifaceted process of modelling with mouth gesture articulation by analogy in particular characteristics of the referents (objects) of the world. For example, Wittgenstein [1] repudiates the idea that the main ground of the meaning is a referential relation between words and objects in the world because there is “no *special access* to the referents other than that linguistic access which for some reason we are not satisfied with” (p. 80).

Following Humboldt [2], the central role in word making should be given to humans, their minds, and their sensuous subjectivity that is transformed into a perfectly attuned sense of articulation. The author's [3] tentative Rigorously Universal Phonosemantic Hypothesis (RUPH) on the phonosemantic structure of words is presented. Phonosemantic analysis of some common terms including animal names is provided that allows comparing of the expressiveness of particular languages and cultures on the basis of universal phonosemantics which appears to be a sort of Mentalese and thus can be used in AI and ICT applications.

1. Advancing the Phonosemantic Hypothesis

The conventional phonosemantic hypothesis which Magnus [4] coins at best sounds as:

In every language of the world, every word containing a given phoneme has some specific element of meaning which is lacking in words not containing that phoneme. In this sense, we can say that every phoneme is meaning-bearing. The meaning that the phoneme bears is rooted in its articulation (p. 166).

The suggested rigorously universal phonosemantic hypotheses (RUPH) assumes:

- (1) the trilateral dialectical unity or the sameness [1,3] (also, the general idea of necessary minimal units for any dialectical analysis by Vygotsky [5, pp. 4–5] of sound, mouth gesture, and meaning (as the reference of a thought) in human speech is embodied in
- (2) the ‘pure’ and ‘quasi-naked’ sense of articulation [2, p. 75]; that is,
- (3) the speaker’s proprioception of the positions of the organs of speech which express
- (4) some primordial, often syncretic, meanings of thought which are always intentional and striving to be ‘truly iconic’ [4, p. 1] to reality.
- (5) Whilst being gestural and thus indexical, the articulation produces speech sounds iconically charged with meaning that symbolizes the content of a thought.
- (6) The sequence of often primarily reduplicated, agglutinated, and then ‘fossilized’ sounds – a word – seems to comprise
- (7) the primeval description of some spatial qualities of the referent of a thought – the object. The sounds here substitute for not a particular object – a referent – but its particular characteristics.
- (8) Such a primordial language – eventual *Mentalese* – is constrained and determined mostly
- (9) by human articulatory abilities, and thus able to express only basic, mostly spatial, features and relationships of referents,
- (10) with iconically relevant proprioception of both mouth gestures by the speaker,
- (11) and distinctive acoustic apprehension of phonosemantically categorized phones by the hearer;
- (12) significantly, such basicallity may be not specifically linguistic and thus universal.

It is thus supposed that human speech as a sequence of utterances has appeared earlier than words which were initially built *ad hoc* by agglutination of meaningful phones with their further fossilization and complete loss of the explicit once primordial meaning.

2. The Preliminary Inventory of Phonosemes as Conceptual Primitives

The hypothesis [3] has attributed meaning to particular sounds using a complex of analytic, mostly contrastive procedures, such as: minimal pairs comparison, transfer of the meaning of one-sound words into multi-sound ones, interjection and mouth gesture designation, proprioception of articulatory organs, cross-lingual comparison and exchange of phones’ meaning between languages, designation of compound units (clusters), and transgressive analysis of findings. Several languages have been used for elaboration and verification of the compiled inventory of *phonosemes*, a newly introduced concept of the elementary bearers of meaning – phones of human speech (in Table 1). The new term designates a phone and its relevant transcribing letters. The

Table 1. The Preliminary Inventory of *Phonosemes*

Phonoseme	Grapheme	Syncretic Intrinsic Meaning, or Conceptual Grasp
/a/	<i>A</i>	Openness, boundlessness, maximum, all around, largeness, outsidedness, vagueness, unconstrained entity
/e/	<i>E</i>	Action, emergence, activation, erection, energy, existence
/i/	<i>I</i>	Narrowness, minimum, tension, localisation, spreading out, penetration, insight
/u/	<i>U</i>	Depth, magnitude, content, interiority, outstrechness, origin, (re)source, bulkiness
/o/	<i>O</i>	(In)closeness, surroundings, orderedness, vicinity, limitation
/y/	<i>Ū</i>	Combines the traits of /i/ and /u/
/œ/	<i>Ö</i>	Combines the properties of /o/ and /e/
/b/	<i>B</i>	Being, existence, realisation, fulfilment, confirmation
/d/	<i>D</i>	Givenness, position, determination
/dʒ/	<i>J, G</i>	(Strong) quick, fast (trans)action, command, affiliation
/f/	<i>F</i>	Being inside, within to, from within
/g/	<i>G</i>	(Strong) contact, cohesion, point at, to, towards, enforcement
/h/	<i>H</i>	Direction, pointing at, to, towards
/k/	<i>K</i>	Contact or interaction (present and future), cohesion, pointing at, to, towards, enforcement
/ks/	<i>X</i>	Conjunction within from
/kv/	<i>QU</i>	Within, towards, conjunction with, penetration, into
/p/	<i>P</i>	Alongside, impulse, pushing, preceding
/s/	<i>S</i>	With, together, with contact (past and present)
/ʃ/	<i>SH</i>	Affiliation, long-term conjunction, contact
/t/	<i>T</i>	Direction, impulse, indication, pointing out (at), to, towards, Fixation, end, finish
/ts/	<i>TS, C</i>	Persuasion, conjunction, markedness
/tʃ/	<i>CH, TCH</i>	Quick, fast (trans)action, command, separation
/v/	<i>V</i>	Inside, within to, towards
/z/	<i>Z</i>	A strengthened version of /s/, with tendency to become without, behind
/ʒ/	<i>ZH</i>	(Strong) permanent contact, interaction, cohesion
/sk/	<i>SK</i>	From within, pertinence of, with
/l/	<i>L</i>	Longevity, outspreadness, length, duration, pointing at
/m/	<i>M</i>	Appropriateness, attribution, possession, satisfaction, saturation
/n/	<i>N</i>	Emergence, availability, birth, enforcement, givenness, presence
/r/	<i>R</i>	Reaction, resistance, response, subjectivity, thingness, reality, reflectivity
/w/	<i>W</i>	Combines traits of /u/ and /v/
/j/	<i>Y</i>	Overcoming obstruents, realisation, fulfilment, squeezing out, entry (in initial position) into and exit (in final position) out of

quite close in meaning and already existing in linguistics, such terms as phoneme and phememe are not applied here because both do not convey intrinsic phonosemantic properties of speech sounds. The former is as a concept of particular structuralized and systematic approach; the latter is overcharged with cultural considerations.

For an example of simple phonosemantic analysis,

the Arabic *ALLAH* means: *A* – everything, all, the greatest external entity; *L* – long, continuous; *H* – to, together; *AL* – all embracing, expanding everywhere; *LA* – extending, stretching out all; *LAH* – pointing out everyone, at all ... (everything).

Similarly, the Russian *BOG* ('god') means: *B* – being/to be; *O* – confining; *G* – definitely, strongly to.

The English *GOD* means: **G** – definitely, strongly to; **O** – confining; **D** – giving/given.

The French *Dieu* means: **D** – giving/given; **I** – immense; **E** – energy; **U** – profoundly/almightily.

3. The Multilingual Phonosemantic Universality and Diversity of Conceptualization

3.1. Animal Names Designation

The cross-lingual analysis of names for several animals provided in thirteen languages shows the possibility of the following integrated conceptions:

- for *cat* as “something intending to be outside, not enclosed, avoiding communication”;
- for *goat* as “something intended being expelled outside”;
- for *dog* as “something determined to secure surroundings very much”;
- for *horse* as “something overcoming/bridging the space”;
- for *wolf* as “something surrounding, obtaining, appointing at a desirable object, or a purpose”.

Such findings correspond well to the characteristics traditionally attributed to these animals by different peoples. As another example of the investigation, the case for *horse* is given in Table 2.

3.2. The Phonosemantic Interpretation of Space, Time and Information

The notion *SPACE* may be interpreted as “within from (***S**) passing (***P**) something large (***A**) with (for ***C** = ***S**), or there with (for ***C** = ***TS** in other reading tradition) [action (***E**) – now silent]”.

The word *TIME* would determine “there/this (***T**) intensity/insight (***I**) possessing (***M**) [action (***E**) – now silent]”.

The term *INFORMATION* would designate “immediately (***I**) available (***N**) for/to (***F**) enclosing (***O**) response (***R**) appropriate (***M**) surroundings (***A**) there (***T**) insight (***I**) encompassing (***O**) available (***N**)”.

4. Phonosemantic Ramifications

The suggested phonosemantic analysis elucidates several fundamental linguistic problems: those of double articulation, language origins and diversity, etymology of words, development of language(s), sound symbolism and word formation. The preliminary analysis reveals, however, many problems and complications in interpretation of the results such as genuine syncretism and inexplicability of the primordial concepts, fuzziness and opaqueness of their phonosemantic descriptions. Nevertheless, words here appear to be compressions of cross-lingual primordial meaning and this can be applied for some formal AI and ICT applications (language processing, lexical corpora et al.) where the system may be used to evaluate equivalence among words in multilingual environments. There might be the possibility of making phonosemantics relevant

Table 2. Naming *Horse*

Language	Animal Name	Phonosemantic Description of Conceptual Grasp
English	<i>horse</i>	To(* <i>H</i>) surrounding (* <i>O</i>) reality (* <i>R</i>) with (* <i>S</i>) energy (* <i>E</i>)
Berber	<i>agmar</i>	Something big (* <i>A</i>) intended (* <i>G</i>) to have (* <i>M</i>) large (* <i>A</i>) reality (* <i>R</i>)
Chinese	<i>ma</i> ³	Possessed (* <i>M</i>) of space (* <i>A</i>)
Finnish	<i>hevonen f</i>	Toward (* <i>H</i>) moving (* <i>E</i>) into (* <i>V</i>) surroundings (* <i>O</i>) available (* <i>N</i>) energy (* <i>E</i>) source (* <i>N</i>)
	<i>ratsu m</i>	Responding (* <i>R</i>) to space (* <i>A</i>) with (* <i>TS</i>) such a force (* <i>U</i>)
French	<i>cheval</i>	Alongside with (* <i>SH</i>) energy (* <i>E</i>) to (* <i>V</i>) space (* <i>A</i>) appointed (* <i>L</i>)
Georgian	<i>tskheni</i>	Together (* <i>TS</i>) with (* <i>KH</i>) energy (* <i>E</i>) source (* <i>N</i>) intended (* <i>I</i>)
German	<i>Pferd</i>	To push (* <i>P</i>) alongside (* <i>F</i>) acting (* <i>E</i>) reality (* <i>R</i>) determined (* <i>D</i>) [Paraphrasing: pushing (<i>die Erde</i>)]
Greek	<i>álogo</i>	Everything (* <i>A</i>) inclined (* <i>L</i>) to encompass (* <i>O</i>), strongly intended (* <i>G</i>) object (* <i>O</i>) [Without logging the space]
Japanese	<i>uma</i>	Much of outside (* <i>U</i>) possessed (* <i>M</i>) large agency (* <i>A</i>)
Lithuanian	<i>arklys</i>	Space (* <i>A</i>) overriding (* <i>R</i>) to (* <i>K</i>) large extend (* <i>L</i>) intended (* <i>Y</i>) with (* <i>S</i>)
Mongolian	<i>mor'</i>	Possessed (* <i>M</i>) of surrounding (* <i>O</i>) reality (* <i>R</i>)
Russian	<i>loshad'</i>	Appointed (* <i>L</i>) surroundings (* <i>O</i>) together with (* <i>SH</i>) all the rest space (* <i>A</i>) to join Intended (* <i>D</i>)
Turkish	<i>at</i>	Spacing (* <i>A</i>) there (* <i>T</i>)

to computers, either in articulation (where there is semantics in the computer output), or as an element of that how the computer acquires speech (unlike current systems which are essentially pure signal processing), both at a lower level (identifying phones) and at a higher level (associating semantics with what is heard). The applicable scope of the proposed system can eventually deal with any sort of words. Thus, it can be used as one of the core parts of machine translation and ontology of a semantic web.

Conclusion

The particular audible characteristics of *phonosemes* – phones of human speech as the elementary bearers of meaning – are determined by articulatory positions of the mouth organs. The analogy in the mouth articulation of words and their referents' traits bridges the explanatory gap between the content of human thought and the surrounding world. The eventual confirmation of the suggested hypothesis on the trilateral unity (conceptual sameness) of mouth gesture, thought, and sounds in human speech helps solving some fundamental linguistic problems, and linguistics gains some monistic and unifying coherent principles. Words appear to be compressions of cross-lingual primordial meaning and this can be applied for some formal AI and ICT applications (language processing, lexical corpora, machine translation, semantic webs, et al.).

References

- [1] Finch, Henry Le Roy. Wittgenstein – The Later Philosophy: An exposition of 'The Philosophical Investigations'. Atlantic Highlands, NJ: Humanities Press, 1977.
- [2] Humboldt, Wilhelm. *On Language: On the Diversity of Human Language Construction and Its Influence on Mental Development of the Human Species*, ed. M. Losonsky, trans. P. Heath. Cambridge: Cambridge University Press, 1999 [1836].

- [3] Medvedev, Alexei. 'Contrastive phonosemantics as a basis for word formation', in *Studies in Contrastive Linguistics: Proceedings of the 4th International Contrastive Linguistics Conference, Santiago de Compostela, September, 2005*, eds C. M. Figueroa and T. I. Moralejo Gárate. Santiago de Compostela: University of Santiago de Compostela, 2006. 611-616.
- [4] Magnus, Margaret H. *What's in a Word: Studies in Phonosemantics*. [Dissertation]. Kirksville, MO: Truman State University Press, 2001. <http://www.trismegistos.com/IconicityInLanguage/Dissertation/> [On-line 02/04/04].
- [5] Vygotsky, L. S. *Thought and Language*, rev. and ed. A. Kozulin. Cambridge, MA: MIT Press, 1986.

Information Modelling for Preference Analysis of Musical Instrument Sounds

Yukari SHIROTA

*Faculty of Economics, Gakushuin University, 1-5-1 Mejiro, Toshima-ku,
Tokyo, 171-8588 Japan*

Abstract. In the paper, I propose a new information modeling for preference analysis of musical instrument sounds. I think that the Gunji's classification method is suitable as the base of the model. Concerning the preference analysis, I am planning to use the conjoint analysis methods and I discuss the problems to overcome while conducting the conjoint analysis on the model.

Keywords. Musical instrument, museum, conjoint analysis, preference tendency, Gunji's systematics

Introduction

One of my areas of particular interest is capturing children's responses to different cultures. It is the case that many people from a Western culture are familiar with Western/European music idioms and sounds, but not particularly familiar with Eastern/Asian idioms and sounds. On the other hand, many people from an Eastern/Asian culture are familiar with Eastern/Asian music idioms and sounds, but not particularly familiar with Western/European idioms and sounds. An interesting study would be on ways of teaching these differences at an early age, by gauging young children's reactions to different musical cultures. A more ethnographic study might be to determine how early in life music cultures are embedded within a child's likes/dislikes. From these reasons I would like to conduct analysis of children's preferences about musical instrument sounds, especially of early age instruments [1]. For the analysis, I think that conjoint analysis is suitable. However, there is no model that is appropriate for the conjoint analysis among the existing musical instruments. Then firstly I have to define the new sound model for the conjoint analysis.

In the next section, I propose the Gunji's systematics as the base of the information modelling. In Section 2, I describe some problems while applying the Gunji's systematics for the conjoint analysis. Then I propose the step-by-step analysis plan to solve the problems. Finally the conclusions are described.

1. Gunji's Systematics of Musical Instruments

In this section, I shall explain attributes to feature musical instrument sounds. I have decided to use Prof. Gunji's systematic methodology to express the musical instrument

sound attributes.* Currently there are too many classification methods about musical instruments around the world [2]. However, I think they are not appropriate as an information model of the musical instrument sounds. The feature of Prof. Gunji's methodology is to consider musical instruments from the perspective of various phenomena of sound [3]. In Prof. Gunji's methodology that we call in short "systematics", a musical instrument is expressed by the physical features about sound generation. Therefore, the classification becomes clear and has no ambiguity.

In the systematics, there are seven attributes as follows:

- (1) Form of Vibrating Body
- (2) Material of Vibrating Body
- (3) Source of Vibration
- (4) Application of Vibration
- (5) Conversion of Vibration
- (6) Form of Converting Part
- (7) Material of Converting part

Then I will discuss the classification/descriptive power of the systematics. First let me consider the keyboard instruments. Given the condition that the fourth attributes [application of vibration] is a mechanical one, then almost all the instruments belong to the keyboard group. Then the keyboard instruments could be divided on the attribute [vibration sources] as follows:

- Percussion: piano/clavichord [5413332], carillon [24131--], glockenspiel [4413215]
- Plucking: harpsichord/spinet/virginal [5433332]
- Air current: reed organ [4443215], positive organ [4543265]

Because my research target is musical instruments of an early age, electronic oscillation ones are removed here. These systematic codes can spot the difference between pianos and harpsichords on the third attribute "source of vibration"; pianos strike strings with hammers and harpsichords pluck strings with plectrums on the jacks. On the other hand, the difference between pianos and clavichords cannot be expressed on the code, although these action mechanisms are quite different; the clavichord produces sounds by striking strings with small metal blades called tangents. The reed organ which belongs to the keyboard instruments is, however, a wind machine as shown in the systematic code.

The mechanization of musical instruments can be expressed by the systematic. Focusing on the attribute [application of vibration], we can see some mechanization examples from simple instruments as follows:

- a) Panpipes [4541265] to Pipe organs [4543265],
- b) Chord Zithers [5431332] to Harpsichords [5433332], and
- c) Dulcimers [5412332] to Pianos [5413332].

Next let me consider the reed instruments, woodwind and brass. Woodwind instruments with single reeds (e.g., clarinets, saxophones) and double reeds (e.g., oboes, bassoons) have the same code [4241265] as far as I know. On the other hand, although bagpipes have single beating reeds in the drones, the code is [4242265]. This is because the players activate the reeds by squeezing a bag filled with air held under their

* Prof. Sumi Gunji is a retired professor of Kunitachi College of Music in Japan.

arms [5]. Then the activation is ‘indirect’. Then let me focus on an air reed of the reed instruments. The wind instruments with air reeds can be expressed by [454****]. Among them, there is a difference between the direct application of vibration [454I***] and the indirect one [4542***]. For example, a recorder is expressed as [4542265] and an ocarina is expressed as [4542215]. On the other hand, in the case of flutes or Japanese shakuhachis, the players use their facial muscles and the shaping of the lips to make the air reeds [5]; there is no air reed in the instrument. Therefore the code becomes [454I265], which means the direct application of vibration. Now let me consider the brass instruments such as trombones, cornets, trumpets, French horns, and tubas. In the brass, the vibration body is the player’s lips. Then the code becomes [4I41265] where the value [I] means the material is ‘a part of the human body’. Then the shape of mouthpieces affects the sound. However the differences cannot be expressed on the systematic code.

2. Conjoint Analysis on the Systematics

In the section, I shall consider the conjoint analysis on musical instrument sounds. The explanation of the conjoint analysis and the detailed discussion were described by Shiota [6]. There are some problems to overcome so that we can apply the conjoint analysis. The problems are summarized as follows:

1. It is difficult to determine a representative instrument for one concept which corresponds to the subgroup of the instruments.
2. It is difficult to determine a representative piece of music/melody (the main tune) played for the concept, because the selected piece of music has effects on the respondent’s evaluation of the sounds.
3. Some problems related to the systematics:
 - The number of attributes and the levels on the systematis is too large to handle while we conduct a conjoint analysis.
 - Some attributes partly depends on the others, although any two attributes must be independent on each other in the conjoint analysis.
 - Sometimes there is no existing musical instrument corresponding to a concept which is generated by the orthogonal main-effects design.

Concerning the second problem, another reason is that we cannot remove the collaboration effects in most cases. Therefore we should try to select right recordings which feature the instrument sound so that the respondent can concentrate on the target sound as much as possible.

The most significant problem to conduct the conjoint analysis was that there was no existing musical instrument corresponding to the concept. To solve the problem, I firstly had to survey the dependency among the main three attributes using the real museum database. Then I had selected the digital catalogue of Gakkigaku Shiryokan (Collection for Organology), Kunitachi College of Music in Tokyo.[†] The reasons why I selected this digital catalogue were that this was well-organized and a Web-published database. Another reason is that it is too difficult for me to make the systematics code and that this catalogue is the only one annotated with code of the systematics.

[†] http://www.gs.kunitachi.ac.jp/e_cat.html.

The number of musical instruments						
	1	2	3	4	5	6
	Solid	Hallow solid	Stick	Board	String	Membrane
1 Percussion	4	272	23	184	71	239
2 Friction	0	15	12	4	135	0
3 Plucking	0	0	0	45	290	0
4 Air current	1	0	0	1024	2	1

Figure 1. Survey result of the dependency among [form of vibrating body] and [source of vibration].

	1	2	3	4	5	6
	Solid	Hallow solid	Stick	Board	String	Membrane
1 Percussion		1: NO 2: MANY 3: NO	1: NO 2: MANY 3: NO	1: NO 2: MANY 3: NO	1 NO 2:Dulcimer 3: MANY	1: MANY 2: MANY 3: NO
2 Friction					1:NO 2: MANY 3:Hurdy-gurdy	
3 Plucking						
4 Air current						

Figure 2. Survey result of dependency among [form of vibrating body], [source of vibration] and [application of vibration].

Figure 1 shows the survey results of dependency among attributes [form of vibrating body] and [source of vibration]. As can be seen there, there are many empty cells. Figure 2 shows another survey result of dependency with the attribute [application of vibration]. From the result, I found that only [Plucking-String] and [Air current-Board] subgroups had existing instruments corresponding to every set of the three attributes. In conclusion of the survey results, we can say that there exist some groups that have no existing musical instruments and that there are some strong dependency relations among the main attributes. Therefore it is impossible to conduct the conjoint analysis if I do not change the current approach.

So I proposed here the step-by-step analysis plan. There the respondent is firstly asked “which instrument family do you like best?” The candidate families are (1) percussion, (2) wind, (3) string-percussion, (4) string-friction, and (5) string-plucking. If (2) wind or (5) string-plucking are selected, then we can conduct the conjoint analysis on the three main attributes [source of vibration], [form of vibrating body], and [application of vibration].

3. Conclusion

In the paper, I proposed a new information modelling of musical instrument sounds suitable for analysis of likes and dislikes about the sounds. I think Gunji’s systematics is suitable as the base of the model because the attributes feature the physical phenom-

ena of the sound generation. In addition, I have discussed the problems when I try to apply the conjoint analysis on that. The most significant problem was that there were no existing musical instruments corresponding to the conjoint concepts. To solve the problem, I have surveyed the dependency among the main three attributes using the real instrument database and proposed the step-by-step analysis plan to avoid the empty parts.

Acknowledgements

I offer my thanks to Dr Matthew Dovey, JISC (UK Joint Information Systems Committee) Programme Director for e-Research, who originally proposed the research theme “preference analysis about musical instrument sounds” and Prof. Sumi Gunji, from whom I have learned a lot concerning musical instruments. I would like express my deepest gratitude to her.

References

- [1] Yukari Shiota: “A Piece of Advice on Development of Virtual Museums of Musical Instruments – Proposals for Educational Uses of Musical Instruments Museums–”, Proc. of Data Engineering Workshop, The Institute of Electronics, Information and Communication Engineers (DE), March 9 to 11, 2008, Miyazaki, Japan, A8-1.
- [2] Wikipedia: “Musical instrument classification”, at http://en.wikipedia.org/wiki/Musical_instrument_classification.
- [3] Sumi Gunji, Gakkigaku Shiryokan (Collection for Organology), Kunitachi College of Music: “Concerning Systematics”, at http://www.gs.kunitachi.ac.jp/e_catleg.html.
- [4] Neville H. Fletcher, Thomas D. Rossing: *The Physics of Musical Instruments (Section 17.1)*, Springer-Verlag, New York, 1998. The Japanese translation by Kenshi Kishi, Hidemi Kubota, and Shigeru Yoshikawa. Springer-Verlag, Tokyo, 2002.
- [5] Neil Ardley: *Eyewitness MUSIC*, Dorling Kindersley, London, 1989.
- [6] Yukari Shiota: “Proposal for Analysis of Likes and Dislikes about Musical Instrument Sounds”, Proc. of Data Engineering Workshop, The Institute of Electronics, Information and Communication Engineers (DE), March 9 to 11, 2008, Miyazaki, Japan, E8-4.

Opera of Meaning: film and music performance with semantic associative search

Shlomo DUBNOV^a and Yasushi KIYOKI^b

^a *Music, California Institute of Telecommunication and Information Technology,
University of California in San Diego, La Jolla, 92093, USA*

^b *Faculty of Environmental Information, KEIO University, Fujisawa, Kanagawa 252,
Japan*

Abstract. Recently artists are exploring ways for incorporating large amounts of information and networking as part of their medium. One of the main challenges in applying information technology to film and opera is in relating different types of media to the meaning of story narrative. Opera of Meaning is a new format for distributed, collaborative and interactive viewing where the association of different media elements is done dynamically by semantic and impression search that is performed by the public during the performance in context of a main story. This opens new research questions in database modeling and semantic technology related to story meaning, media auto-tagging, automatic editing and mixing, user interaction, social networking and more. We plan to offer this format to artists, producers and the public, opening a new venue for social creation and experiencing of impression and meaning in digital media.

Keywords. Opera, Film, Kansei, Expectancy, Mathematical Model of Meaning

1. Introduction

Opera of Meaning (OoM) is a format of presentation of media contents, such as dramatic work, entertainment or lecture, that uses a realization of a script (main story) together with a collection of related clips and other electronic information (related contents) which are dynamically presented in parallel with the main story during the show. A central feature of Opera of Meaning is that the public can interact with related contents during the story presentation by providing information to the performance system regarding selection of related contents and by posting their own information on designated publicly viewed display areas. The main story can be live, prerecorded or a combination of the two, such as TV show, music performance, film screening or other types of electronic transmission. In a recent piece based on an ancient historical text [1][2] a live performance occurred on a central stage simultaneously with projection of films on a series of screens surrounding the audience. At dedicated times the live performance was stopped in order to present films and video interviews with experts about the story and to conduct public discussion by having the audience present their own contents such

as text and graphics on the surrounding screens, and express themselves via electronic chats and votes on questions related to the story. The name “Opera of Meaning” is intended to emphasize the interplay of meanings and exchange of opinions that occur during performance of a story. This approach is related to works that explore social and educational roles of film and theater media, and is inspired by methods of debate and commentary that are common in traditional religious study situations such as Jewish Talmud or Tibetan Buddhism.

1.1. Relation between information technology and story telling

Storytelling is the ancient art of conveying events in words, images, and sounds often by improvisation or embellishment. When a story is presented in a dramatic manner in one or more acts set to music and artistic visual expression, it becomes film or opera. Today, many methods are used to introduce more sophisticated structures and additional information into traditional stories. Simultaneity, split screens and sophisticated editing become inherent part of digital aesthetics. Metafiction and the use of a commentator allow embedding stories inside or in parallel to other stories, dealing with relation between fiction and reality. Information technology introduces yet more new and unexpected ways to tell and understand stories. In OoM the use of related materials presents an opportunity to “embed” user defined stories in the original plot. The commentary method, besides its obvious function as a learning tool, is understood as a shift in attention of the readers from plot to characterization. Building upon contemporary culture of social “tagging”, OoM offers additional characterizations of main story in ways that are very different from traditional hyperlink approaches. In addition, public interaction aspects are designed in a manner that maintains dynamic flow and coherence of presentation, inspired by methods of machine improvisation [3] that use string matching over database of music materials to retrieve, recombine and create music.

2. System Concept and Architecture

Our system combines concepts of Kansei, narrative, commentary and debate. In the field of multimedia database systems, the concept of Kansei is related to data definition and data retrieval with impression information for multi-media data, such as images, music and video. The multimedia data, together with its semantic and impression descriptors are used to define the “information world” in which the user can operate that is relevant to a particular story. The closed-world-assumption in database modeling is suited to the Opera performance, because we can prepare appropriate information resources for it in advance not from open-world (WWW) but from closed-world. According to our OoM concept, the DB design and story annotation processes are included in the scenario design.

2.1. The Opera performance system

The main principles behind designing the opera performance system are:

1. presentation design that establishes clear relations between the main story and related contents

2. timeline organization according to different levels of viewing and user interaction activities, such as performance acts, commentary and debate.
3. close world database and logical structure of the story are included in the scenario design
4. public participation is allowed according to reactive, deliberative and reflective interventions on informative and kansei levels

Design of the presentation system is done according to a principle of separation between main story and related contents in a manner that maintains the centrality of the main story and by provision of surrounding materials in a spatially and temporally coherent manner. By careful control of the amount, rate and type of information that is allowed to be displayed from related contents, the total sum of audiovisual materials enhances the audience experience by double coding and by engagement through active participation. The architecture of the theater version of the system is shown in Figure 1.

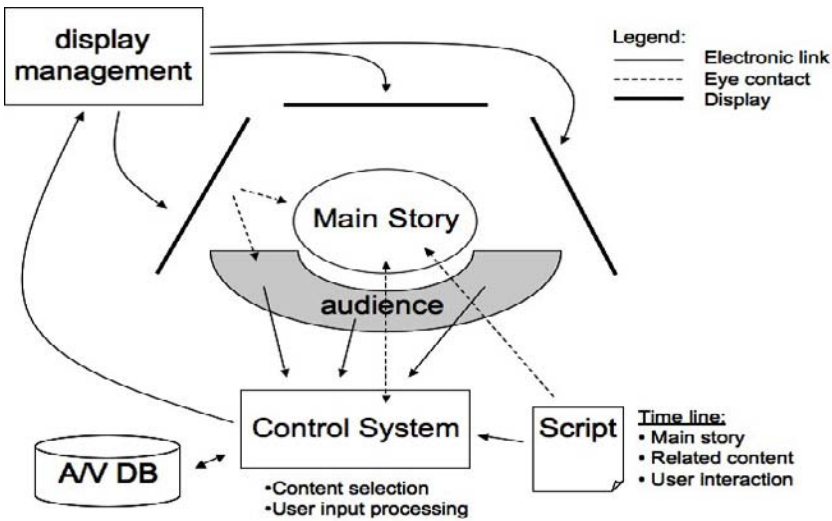


Figure 1. Performance system architecture

2.2. Multidatabase system architecture

In order to support various types of data (different story scenes, logic of story structure, rules for content composition and presentation, databases of related text, images, video), a meta-level active database functionality is used to integrate among heterogeneous resources. There are three main operators that retrieve the relevant content related to the story scene: The semantic component matches informative contents such as specific keywords related to story contents. Impression components operate on the kansei level and “filter” the data according to mood, emotions and other aesthetic criteria. The story / world component considers the role of data in story presentation such as identifying functions related to point of view, chronological and spatial editing considerations and keeping track of expectations and questions derived by the audience in response to story

segments. This architecture is presented in Figure 2. The database system interfaces the opera performance system through a set of actions defined by the script and user queries.

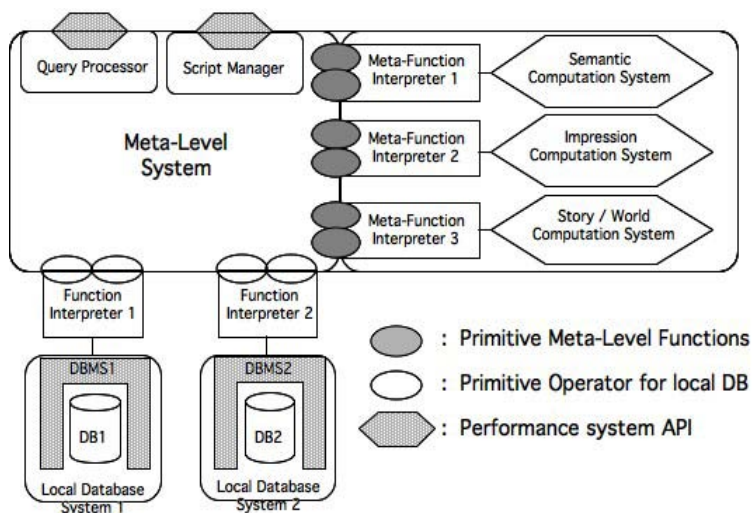


Figure 2. Multidatabase system architecture

3. The Mathematical Model of Meaning for Semantic Associative Search

In the design of the OoM system, one of the important issues is how to deal with “Kansei” of humans. The concept of “Kansei” includes several meanings on sensitive recognition, such as “impression”, “human senses”, “feelings”, “sensitivity”, “psychological reaction” and “physiological reaction”. In the OoM system, the concept of Kansei is related to data definition and data retrieval with Kansei information for multimedia data, such as images, music, video and stories. The important subject is to retrieve images, music, video and stories dynamically according to the user’s impression given as Kansei information. The field of Kansei was originally introduced as the word “aesthetics” by Baumgarten in 1750. The aesthetica of Baumgarten had been established and succeeded by Kant with his ideological aesthetics. In the research field of multimedia database systems, it is becoming important to deal with Kansei information for human beings for defining and extracting media data according to impressions and senses of individual users.

As one of the database systems dealing with Kansei information, a semantic associative search system based on the Mathematical Model of Meaning (MMM) has been proposed [4][5][6][7]. The MMM realizes media data retrieval by receiving keywords representing the user’s impression and the media data contents. The MMM provides semantic functions for computing specific meanings of keywords which are used for retrieving media data unambiguously and dynamically. The main feature of this model is that the semantic associative search is performed in the orthogonal semantic space. This

space is created for dynamically computing semantic equivalence or similarity between the metadata items of media data and keywords.

The basic principle in the MMM is that each media data item, which can be text, image, animation, music, or movie, includes various meanings. That is, the meaning of a media data item is not fixed statically. The meaning of a media data item is fixed only when we know the context for explaining the content of a media data item. The MMM defines semantic functions for performing the semantic interpretation of a content and for selecting semantically related media data items, according to the given context. In MMM, metadata expressed in terms of English words are assigned to each media data item.

In MMM, data items are mapped onto an orthogonal space. Each media data item is placed as a single coordinate point in the space and dynamically extracted by semantic associative search. In this semantic space, with approximately 2000 dimensions in current implementation, each context corresponds to one of the subspaces. The subspace is named “semantic subspace.” When the context is given, a semantic subspace corresponding to the context is selected. This selection reflects the recognition of the context given as an aspect. Each data item is also mapped onto the semantic subspace selected according to a given context, and the relationships between data items are dynamically computed by using the metric in the selected semantic subspace reflecting the context. In MMM, the number of phases of contexts is almost infinite, currently approximately 22000.

3.1. The Outline of the Mathematical Model of Meaning

In this section, we briefly review the outline of the semantic associative search method which is based on the Mathematical Model of Meaning. The model has been presented in [4][5][6][7] in detail. The semantic associative search method consists of three steps as follows:

STEP1: Creation of the metadata space:

The semantic associative search for information resources is realized by the mathematical model of meaning. A metadata space is created as a basis for computing the relationships between data items. When m data items are given as the basic data items for creating the space, each data item is characterized by n features. The m basic data items is given in the form of an $m \times n$ matrix M . Computing the eigenvalue decomposition of the correlation matrix $M^T M$, an orthogonal semantic space is created (M^T represents the transpose of M). It is defined as the metadata space \mathcal{MDS} .

STEP2: Mapping data onto the semantic space:

A set of keywords as a query and target data items, both of which are characterized by the same n features as used in STEP 1, are mapped as vectors onto the metadata space \mathcal{MDS} . The MMM measures the association or correlation between context words and each candidate data item. Suppose a sequence of associated context words is given to search a data item, (e.g. peaceful, silent). We can regard the context words as those which form the context. The context is used to select the subspace from the metadata space \mathcal{MDS} .

STEP3: Semantic associative search:

First, when a context for explaining the meaning of a query is given, then the semantic subspace is dynamically extracted from the metadata space. In this model, each context given by a user corresponds to one of the semantic subspaces. Second, target data items are mapped onto the subspace. Then, by calculating the correlation of each data items in the selected subspace, the data items which are highly related to the given context can be extracted. Since the subspace reflects the given context words, the norm of the data item projected onto the selected subspace is regarded as the correlation between the data item and the given context words. That is, the data item with a larger norm is highly related to the given context, and is obtained as the appropriate data item for the context. The data items with higher norms are obtained as the resultant sets which are highly related to the given context.

4. Production of story in OoM format

Production of OoM content consist of script writing / score composition, context annotation, related content collection, collage and montage design, and formulation of public interaction rules during performance, commentary and debate phases. This is similar to the process that was used to create the Kamza and Bar Kamza performance [1][2]. The starting point in this process is a textual script that must be rendered into a musical and visual depiction of the story. This text is extended into a multiple-track script that is described in the next paragraph. The next step is collection of related content and its association with the main story. The goal of related content is expanding the story into additional meaning and impression domains by creating associative links to other sources of information. These associations are developed during deep analysis phase and they may include aspects of literary criticism, social analysis, historical, economical, political, psychological, emotional, religious aspects and so on, depending on the specific story and the public who wishes to participate in the collaborative production.

4.1. Script format and method

The scripting format for the Opera is used to define the variable contexts and consists of the following:

Scenario: A time line describing the main story scenes, related content display and public interaction activities. The complete story consists of multiple acts, divided into performance, commentary and debate.

Score: For each story scene the script provides performance instructions written in multiple simultaneous lines resembling a musical score. Each line in the script provides metadata for association of user queries with story context and instructions for audio mixing and visual display layout. It also provides rules and authorizations for audience participation during scenes.

Tracks: The different elements contributing to the final display consist of the main story and related media that are retrieved dynamically during performance. In live version of the Opera, the main story consist of text and music score provided to the actors or musicians. In the film version of the opera system, the main story is represented by a pre-recorded media, such as film, slide show, text or audio narration. Related contents

consist of additional images, video clips, text, graphics and possibly sounds that are selected from a database.

Conductor / Editor: This is part of audio-visual management system (also considered as control room) that is in charge of coordinating the different tracks according to instructions provided in the score. Since many details of the final presentation are decided "on the fly", the score is in fact a type of structured improvisation, and the conductor's role is to monitor the overall result and provide editorial decisions. The Opera system requires careful composition of the overall visual and sonic elements so that the overall effect of the combination between main story and related contents will achieve a certain level of coherence and engagement. The main and related contents must be not only semantically but also aesthetically organized in term of Kansei relations, spatial arrangement and temporal structure.

5. Presentation systems

Presentation design aspect of the OoM system refers to decision about placement of the main story, audience, graphical user interface (GUI) and related contents displays in physical or remote environment. A separate audio design is done to provide an optimal sound reinforcement and sound surround effects. Two examples of presentation design that are presented below are a shared common space called "hyper-cinema theater" and a personal multi-display viewing system.

5.1. Hypercinema theater

The principal goal for the hypercinema theater is to provide a space for public presentation of works that integrate immersive and multi-layered video projections and audio reinforcement for film or mixed live and cinematic productions. The design elements include lighting, scenery, and video projection on two, or more screens. Additional aspects of the design includes division of the physical theater space into areas of main story performance and audience presence. Seating of audience can be done in several configurations relative to the main story performance area. In all configurations the audience is surrounded or flanked by projection screens that display the related contents. The main story area can be used for mounting a film projection display or as a stage for live performance. The audience area can also be reinforced with microphones that are fed into the audio system. Lighting is used to transition between the different acts in the performance, through projections, and debate sections when present. A local wireless network is used to access and conduct communication between the audience and the performance system, such as posting display requests, conducting chats or submitting votes during the different performance.

5.2. Personal multi-display system

A personal display system is used for viewing of pre-recorded content with personal interaction. It consists of a single or multiple displays that are logically and functionally divided into three areas corresponding to main story, related contents and GUI. For instance, a graphics expansion module could be used to extend the desktop across multiple

screens. In such a case a custom software must be provided that divides the two screens into the three functional areas. One option is to split the first display into main story player and GUI, and devote the second screen solely to related contents. Another option is to use the first display for GUI only and divide the second display in software between main story and related content. In a later version of the system a networked shared viewing will be developed. This requires a synchronized streaming of audiovisual contents to multiple users and communication between the users and the server.

6. Applications

The system will be used to create content and produce events in OoM format. In Kamza and Bar Kamza story from the Talmud, a recording of a stage performance will be used as the main media and collections of images, maps, text and movies will be stored in the database. Another original story "Gadget" (in preparation) is an adaptation of Far Eastern folk tale "Mirror" that presents social criticism using a humoristic plot [9].

Acknowledgements

This work is partially funded by Heiwa-Nakajima-Zaidan fellowship. Debate and Commentary Play project is supported by California Institute of Telecommunications and Information Technology and the UCSD Chancellor's Interdisciplinary Collaboratories program.

References

- [1] D. Ramsey and D. Sutro, "An Opera of Meaning Integrates Live Performance, Internet, Multimedia and Audience Participation at UC San Diego", UCSD News, February 2008, <http://ucsdnews.ucsd.edu/newsrel/arts/02-08OperaOfMeaning.asp>
- [2] Debate and Commentary Play, <http://kamzaandbarkamza.wikidot.com>
- [3] S. Dubnov and G. Assayag, "Memex and Composer Duets: computer-aided composition using style mixing", Open Music Composers Book 2, Collection Musique/Science, Editions DELATOUR France, 2008
- [4] Y. Kiyoki, T. Kitagawa and T. Hayama, "A metadatabase system for semantic image search by a mathematical model of meaning", ACM SIGMOD Record, Vol.23, No. 4, pp.34-41, Dec. 1994.
- [5] Y. Kiyoki, T. Kitagawa and Y. Hitomi, "A fundamental framework for realizing semantic interoperability in a multidatabase environment," Journal of Integrated Computer-Aided Engineering, Vol.2, No.1(Special Issue on Multidatabase and Interoperable Systems), pp.3-20, John Wiley & Sons, Jan. 1995.
- [6] Y. Kiyoki, A. Miyagawa, and T. Kitagawa, "A multiple view mechanism with semantic learning for multidatabase environments," Information Modelling and Knowledge Bases (IOS Press), Vol. IX, May, 1998.
- [7] Y. Kiyoki and T. Kitagawa, and T. Hayama, "A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning," Multimedia Data Management – using metadata to integrate and apply digital media –, McGrawHill(book), A. Sheth and W. Klas(editors), Chapter 7, March 1998.
- [8] Y. Sato and Y. Kiyoki, "A semantic associative search method for media data with a story," Proceedings of the 18th IASTED International Conference on Applied Informatics, pp., Feb., 2000.
- [9] J.H.Grayson, "They First Saw a Mirror: a Korean folktale as a form of social criticism", JRAS, Series 3, 16, 3 (2006), pp. 261-277

Intelligence and Language – How Could Human Being Have Language ?

Setsuo OHSUGA

Professor Emeritus, University of Tokyo

Abstract. Origin of language is discussed. Language is a media to represent, process and transfer concepts. Before language has been made human being has only biological system to preserve their lives like the other animals. How was concepts made and language originated from there? In this paper a possibility of language having been originated is discussed.

Keywords. Intelligence, Language Origination, Neural Network, Concept Formation, Language Model

1. Introduction

The objective of this paper is to discuss a way language has been originated. The final goal is to analyze the progress of human intelligence. Language origination was its most important step.

It is difficult to define intelligence because of its philosophical nature. But apart from its philosophical aspect, to analyze its functional aspect is useful for its understanding. In this point of view, one can see easily a close relation among intelligence, knowledge and language, and define a functional aspect of intelligence.

Knowledge is well defined. It is approved concepts of facts and ideas that were represented formally in language. Usually multi-level knowledge is represented by means of multi-level representation formalism of language. Higher level knowledge has deeper meaning than lower level knowledge and are more useful.

On the other hand, the major role of language is to represent thing(s) and to process them. Its function is smooth communication between persons and problem solving based on knowledge. Deductive inference plays an important role there. It generates new knowledge representation in the same level as existing knowledge and, as the results, expand the scope of knowledge.

Deductive inference itself is not intelligence because it is involved in language and anyone can use it without knowing its deep meaning if he/she can use language. On the other hand, if new knowledge is added that has not been in the existing knowledge base nor in its expansion by inference, then the scope of knowledge expands considerably. It may be agreeable to define intelligence as this function to create knowledge.

Thus language defines a function of intelligence via knowledge and also plays an important role in performing this function. Typical methods of making knowledge are,

- (1) To create new concept (new knowledge) from old concepts [1],
- (2) To discover new knowledge from data [2].

These functions are performed by making use of language and new knowledge that was obtained in these methods is added to the old knowledge.

But what is language? This is another difficult problem to answer. Existing language is a complex entity and was analyzed by many notable philosophers such as G. W. Leibniz, W. von Humboldt, E.G.A. Husserl, F.L.G. Frege, F. de Saussure, B.A.W. Russell, L.J.J. Wittgenstein, etc. But what was the beginning of language and who could make it? It is sure that human being had no language at the outset of history but had it at some point in it. Did human being create language?

One of the important functions of language is to communicate with the others. It is to exchange concepts. Before language, every human had to create concept by him-/herself while after language most people accept concepts from the others by means of language and then comprehend them. It helped human being to accumulate knowledge and accelerates the evolution of intelligence. In other words, intelligence level of human being was very low. How could human being with such limited intelligence create language? It seems a mystery. Recently people who have interests in the origin are increasing [3]. The author also discusses the origin of language. The originality of the paper is to discuss a possibility of language having been originated from the biological system in the brain of primitive human being [4].

2. Biological Processing

2.1. Neural Network as Biological Processing

There are two different kinds of processing in human brain. One is biological processing and the other is language processing. In reality biological processing was only actual processing. The basic biological processor in living things is neural network (NN hereafter) composed of neurons. Human brain is composed of NNs and it became complex by the increase of neurons by evolution and development. Now everyone has about 100 Billion neurons and 100 to 1,000 trillion synapses. In the brain of every person neurons grow very rapidly after the birth.

In early time NNs were only for preserving life. There are two types of neural systems; autonomous nervous system, e.g. maintain temperature, and non-autonomous system, e.g. getting food and crossbreeding intentionally etc.

There are different functions by NNs such as for sensing external signal, for driving internal actuator, for controlling systems by connecting sensor and actuator. But behaviors of many neurons, even their existence, are not well known yet because to analyze it is difficult tasks. For example, people expect as the matter of course that there must be a mechanism to keep temperature of human body. But a path from temperature sensor in skin to heat generation actuator via body heat control (nerve) center in brain was not known but discovered very recently [5]. Bottom-up method like this is the standard approach in neuroscience. But another approach is sometimes necessary for achieving long distance goal such as making clear the human intelligence.

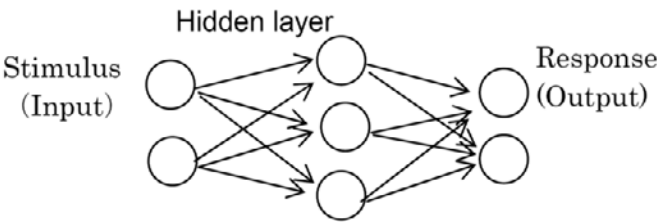


Figure 1. Artificial neural network.

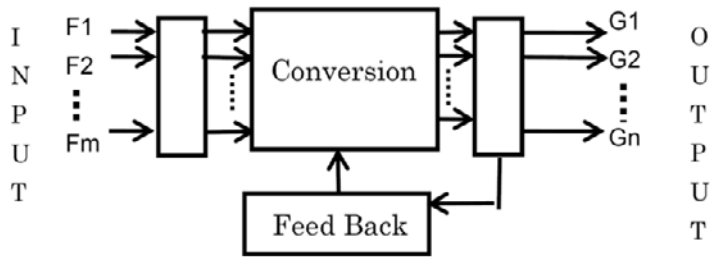


Figure 2. Formal representation of artificial NN.

2.2. Basic Unit Of NN System

A basic unit of NN system is composed of a triple; a sensor NN connected to a control NN and further connected to actuator NN. Internal (microscopic) mechanism of each NN is complex and different by the case. External (macroscopic) input-output specification of NN however is relatively simple. An artificial NN is used as its engineering or mathematical model (Figs 1 and 2). In this paper the scheme as shown in Fig. 2 is used for explanation.

2.3. Growing and Learning of NN

Two typical characteristics of neuron are growing and learning. Neurons grow rapidly after a human was born. Then various functions are implemented to NN by learning. If these functions are well suited for living environment, then the living thing can survive.

Jari Vaario [6] made an interesting simulation of a growth and learning model of artificial life. It was assumed that sensor and actuator has been existed in a creature. The objective of the simulation is to show that a new NN can be created between these sensor and actuator so that the creature can achieve as a whole such actions as getting food and crossbreeding. Growth model was made using LindenMayer model [7] and learning was made. LindenMayer model is a growth model of tree-like structure. It was shown that after many trials it succeeded to achieve the goal.

2.4. Control for Firing NN

When there are more than one NNs, a certain relation must be kept between them for achieving a meaningful operation as a whole. NN is itself a static structure. It is inac-

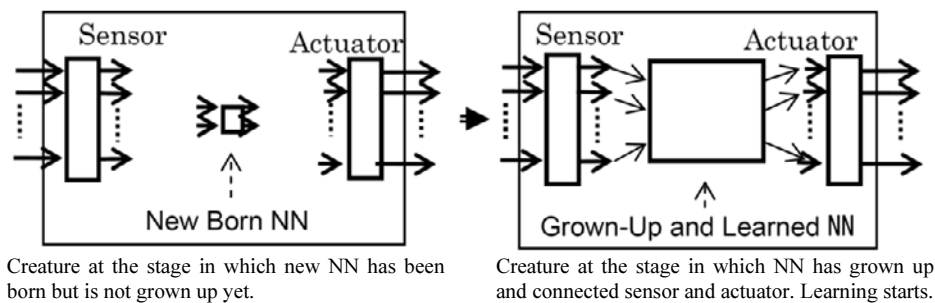


Figure 3. NN growth model.

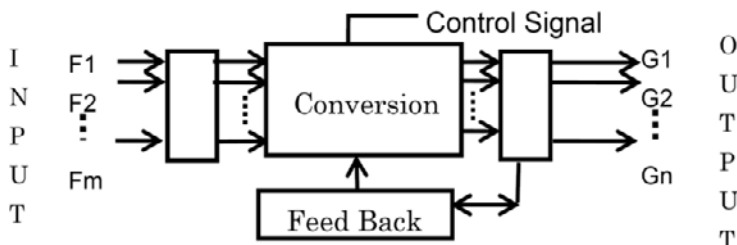


Figure 4. Required Artificial Neural Network.

tive until fired. Every NN must be fired by control signal from outside. It is quite natural to assume that in primitive living thing NN is fired by demand. Living thing has demand generator. NN must have a special port to receive control signal besides ordinary port for receiving working data. That is, there must be two kinds of information in NN; task information and control information as shown in Fig. 4.

It is known that there is special neuron called inhibitory neuron. It stops operation of neuron. Its characteristic is not known well but it surely behaves as a control information. It is assumed here that NN as is shown in Fig. 4 can be the basic component of all biological systems.

3. Notions on Language Processing Based on Biological Processing

3.1. Premises on Discussing Progress of Language

Language was created up-to 30,000 to 50,000 years ago. The way of representation and processing in NN is simple while that of language is the more abstract. But, at an early time, there was only NN system for preserving life. Primitive language was originated there-from. At the beginning it had to be processed by NN. The objectives is to find the way by which language is realized by NN. It is to show that NN has language characteristics. There is no direct key for finding the way. Instead an approach to make a model to represent the process is taken in the following. Before to begin with making such a model the followings notions and assumptions are made.

1. All exiting natural languages (Japanese, English, etc.) have the same form-meaning structure. Therefore a common form is used as a standard to represent every language. Predicate logic is used as common language in the following. A typed logic is used. In the ordinary first order logic, “man is mortal” is written as $(\forall x)[\text{man}(x) \rightarrow \text{mortal}(x)]$. Instead it is written as $(\forall x/\text{MAN})\text{mortal}(x)$ in the typed logic introducing the concept of set. In this expression MAN is a domain set of x . Then, “Socrates is mortal” because $\text{Socrates} \in \text{MAN}$ set theoretically. Also “Japanese are mortal” because $\text{Japanese} \subset \text{MAN}$. In general, let $D; \{D_1, D_2, \dots, D_n\}, D \supseteq D_i$, then $(\forall x/D) \text{predicatel}(x)$ implies $(\forall x/D_i) \text{predicatel}(x)$, $(D_i \subset D)$. Various language (Japanese, English, French, etc.) is translated into this logic.
2. There were two kinds of language in the history of progress of language; holistic language and compositional language [8]. Holistic language is non-compositional but holistic expression to represent meaning while in compositional language sentence is made of a set of words by composition rules. Today compositional language is established and used as standard language.
3. The primitive language was the voice language in which an inner concept is directly connected to voice generation mechanism. Afterward, it extended to symbolic language in which every concept is represented by a symbol.

There were controversies on the type of primitive language [9]. This paper assume that primitive language was holistic voice language and it shifted to compositional language as language has evolved. It is because the narrower the gap between two successive stages of progress of a language is, the larger the possibility of arriving the goal.

3.2. Steps to Arrive at Final Goal

The path to the final goal is decomposed into the shorter paths connecting sub-goals to answer the following sub-problems.

- (1) What is concept?
- (2) How was concept made?
- (3) How was concept connected to holistic language?
- (4) How was holistic language extended to compositional language?
- (5) How did compositional voice language shift to symbolic language?

3.2.1. What Is Concept?

One of the roles of language is for enabling communication. The objective of communication is to exchange concept. From language point of view, concept is what one wants to express in language. It means that primitive concept existed before language and language was made in order to represent the concept. Concept is classified into primitive concept and compound concept. Primitive concept is what cannot be decomposed further. Before language became available every concept had to be represented in the form of physical object (e.g. NN). Compound concept is what is composed from existing concepts. A new (compound) concept was made in language using existing concepts. Compounding of concepts is still continuing today.

It is thought that primitive language was made within the scope of primitive concepts. Various concepts are processed in NN. These are either (1) entity or (2) property of entity or (3) relation of entities or (4) function (behavior) of entity(s). These are what relate closely to human life or activity. An entity is detected by a sensor, a property of an entity is designated by a specific sensor that detected the entity (the sensor can be an identifier of this property), a relation of entities is detected as a co-appearance of entities and is designated by the sensors that detected the entities. But how is the concept of function (behavior) represented by NN? It is discussed in 4.

3.2.2. How Was Concept Made?

Such concepts as entity, property, relation and function (behavior) are made naturally with respect to NN operation. Biological processing is one of the forms to deal with those concepts. Language is another form assuring an easier way of combining concepts. It is sure that primitive language was made on the basis of biological processing.

It was not difficult to define the concepts of entity, property and relation by biological system because these definitions are simple, that is, these are represented by the input/output ports of corresponding NNs. But a concept of function (behavior) is not easily defined like those because it is defined by the operation of NN itself. Biological processing and language processing are very different and the equality of the concept of function (behavior) between two different processing schemes cannot be assured. It is necessary to know the difference between biological processing and language processing first, and then to know if there is something in common between them. For the purpose a framework to compare them is made and these are compared in 4.

3.2.3. How Was Concept Connected to Holistic Language?

Concept was made in a living thing and was uttered in a voice language. A special biological mechanism was used for making concept and uttering it. What are them and how did these biological mechanisms are connected to each other so that the concept is uttered? This is a first issue of language origination and discussed in 5.1.

3.2.4. How Was Holistic Language Extended to Compositional Language?

Number of descriptive expressions by holistic language is limited because number of different voices that can be discriminated is limited. It depends on the mechanism of utterance. It could not satisfy the requirement for increasing concepts as the living thing emerged. It is thought that a number of structures of concepts were made in a living thing and correspondingly the structure were made in language. It is discussed in 5.2.

3.2.5. How Did Language Extend to Symbolic Language?

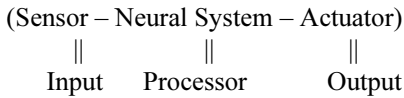
As the structure of language became more complex the language moved to purely symbolic language. It required further development of the functions of biological processing. The mechanism of biological system to cope with symbolic language is not yet made clear. It is mentioned in 5.3.

4. Relation Between Biological Processing and Language Processing

4.1. Framework to Compare Biological and Language Processing

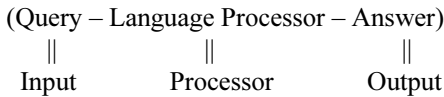
In the early time language was processed by NN. But the primary role of NN was for preserving individual life. It is different from language processing. How was language generated by the NN? It is first necessary to analyze precisely the difference between processing by NN and language processing. For the purpose it is necessary to make a framework to compare them [10].

Biological processing is represented in the form of



The processor generates output by processing input by means of the structure as was shown in Fig. 4. This processing is direct operation to input. Let the input vector be $\mathbf{P}\mathbf{f}$ and the output vector be $\mathbf{P}\mathbf{g}$. Then $\mathbf{P}\mathbf{g} = \mathbf{P}\mathbf{f} \times \mathbf{T}$ where \mathbf{T} is a conversion matrix to represent the scheme of Fig. 4 mathematically.

On the other hand, let the language processing be represented in the similar form as,



The processor generates output by processing input. The structure of language processing is an inference, $(\forall x/D)F(x) \ \& \ (\forall x/D)[F(x) \rightarrow G(x)] \Rightarrow (\forall x/D) G(x)$. It plays an important role in language in both communication and concept creation.

$(\forall x/D)$ denotes common property of all individuals. Let D be a finite set (a,b,c,\dots,z) . Then $(\forall x/D)$ denotes a concurrent processing of all individuals $D = (a,b,c,\dots,z)$. That D is a finite set means to deal with a set of propositions instead of predicate logic. The concept of variable could not be coped with in an early stage of language processing by NN.

$F(x)$ (and $G(x)$) is interpreted as property of x , that is, ' $F(x)$; an element x in D has a property F '. Then $(\forall x/D) F(x)$ presents a state of D with respect to F such that D is in a state in which every element x is $F(x)$. $(\forall x/D)[F(x) \rightarrow G(x)]$ can be interpreted to represent a function or a behavior of the subject of this operation.

In order to represent both biological processing and language processing in the same form, try to interpret deduction as transition such as to obtain $(\forall x/D)G(x)$ from $(\forall x/D)F(x)$ and $(\forall x/D)[F(x) \rightarrow G(x)]$ where input is $(\forall x/D)F(x)$, output is $(\forall x/D)G(x)$ and $(\forall x/D)[F(x) \rightarrow G(x)]$ represents a conversion.

As mentioned above $(\forall x/D)F(x)$ presents a state of D with respect to F . It is possible to define a state space of D of which $(\forall x/D) F(x)$ is a special one. In general a state of D w.r.t F is a combination of $F(x)$ for all x in D . For example, a state such as ' $F(a) : \text{True}$ ', ' $F(b):\text{False}$ ', ' $F(c):\text{False}$ ', ..., ' $F(z):\text{True}$ ' can occur. It is written as $\text{SFI} = (F(a), -F(b), -F(c), \dots, F(z))$. There is $N = 2n (=2^{**}n)$ different states.

Let ' $F(x):\text{True}$ ' be 1, ' $F(x):\text{False}$ ' be 0 and 1 be a binary number $I = 100 - 1$ obtained by concatenating 0 or 1 in the order of arrangement. Then SFI is represented

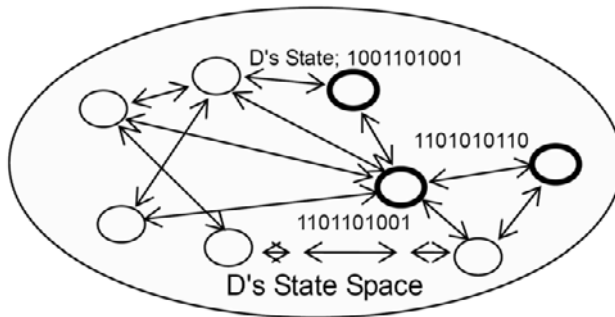


Figure 5. State transition model of logical inference.

$SFI = (1, 0, 0, \dots, 1)$. Among all states, $SF\forall = \{(1, 1, \dots, 1)\} = (\forall x/D)F(x)$ and $SF\exists = \{Sf - (0, 0, \dots, 0)\} = (\exists x/D)$ are the special states. These are the standard states that are defined in the ordinary first order logic.

Next, a state vector $\mathbf{SF} = (SF_0, SF_1, \dots, SF_{N-1})$ and its probability vector \mathbf{PF} are defined. \mathbf{SF} is a state vector in which elements are linearly arranged based on the index I .

Let's assume that truth/false of F for an element can change. When truth/false of at least one element changes, the state of D changes. Let PFI be probability of D being in the state SFI and \mathbf{PF} be a corresponding probability vector. That is, $\mathbf{PF} = (PF_0, PF_1, \dots, PF_{N-1})$. Then it is possible to represent the inference operation as above like $\mathbf{PG} = \mathbf{PF} \times \mathbf{T}$ introducing a transition matrix \mathbf{T} for input-output relation formally. This is the same form as biological processor. That is, there can be a NN represented as $\mathbf{PG} = \mathbf{PF} \times \mathbf{T}$. Note that this does not represent a realistic operation but a virtual relation. It must be finally translated back to a real NN.

4.2. Biological Processor Behaves Like Language Processor Under a Certain Condition

The relation $\mathbf{PG} = \mathbf{PF} \times \mathbf{T}$ mentioned above must meet a special condition, called the condition C1, coming from the relation $F \wedge [F \rightarrow G] \Rightarrow G$. This condition appears in the matrix \mathbf{T} . Namely it must be of a form as shown in Fig. 6. Let it be L-matrix. It depends on the logical expression. Figure 6 is the case of it being $(\forall x/D)[F(x) \rightarrow G(x)]$, $D = (a_1, a_2, a_3, a_4)$.

Condition C1 is made as follows. By definition, $F \rightarrow G = \neg F \vee G$, that is, if $F(x)$ is true for x then $G(x)$ must be true. It means that there is no transition from state SFI including ' $F(x); \text{True}$ ' to state SGJ including ' $G(x); \text{False}$ ' and the element of matrix t_{IJ} for this pair is put zero. As is seen here many elements in this matrix must be zero.

$\mathbf{PG} = \mathbf{PF} \times \mathbf{T}$ is the same relation as NN operation and therefore there can be a NN corresponding with it. This NN behaves like a logical inference $F \wedge [F \rightarrow G] \Rightarrow G$. In other words, biological information processed by NN meeting condition C1 has the same characteristic as language processing. There is an L-matrix corresponding to every different logical form.

L-matrix has many zero elements. There is no such restriction in general biological system. This is the difference between logical inference and biological processing.

	PG0	PG1	PG2	PG3	PG4	PG5	PG6	PG7	PG8	PG9	PGa	PGb	PGc	PGd	PGe	PGf
PF0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
PF1	o	x	0	x	0	x	0	x	0	x	0	x	0	x	0	x
PF2	0	0	x	x	0	0	x	x	0	0	x	x	0	0	x	x
PF3	0	0	0	x	0	0	0	x	0	0	0	x	0	0	0	x
PF4	0	0	0	0	x	x	x	x	0	0	0	0	x	x	x	x
PF5	0	0	0	0	0	x	0	x	0	0	0	0	0	x	0	x
PF6	0	0	0	0	0	0	x	x	0	0	0	0	0	0	x	x
PF7	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	x
PF8	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
PF9	0	0	0	0	0	0	0	0	0	x	0	x	0	x	0	x
PFa	0	0	0	0	0	0	0	0	0	0	x	x	0	0	x	x
PFb	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0	0
PFc	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x
PFd	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0	x
PFe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
PFf	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Figure 6. L-matrix of a logical expression $(Ax/D)[F(x) \rightarrow G(x)]$ $D = (a1, a2, a3, a4)$.

Since inference is a typical characteristic of language, this is very the difference between language and biological systems.

4.3. Characteristic of L-matrix

What is the characteristic of L-matrix? In order to know it let's assume to apply this matrix repeatedly, that is, to make $\lim_{n \rightarrow \infty} T^n$. Then every element of output vector except PGf tend to 0 and PGf tends to1 where PGf is the probability corresponding to the last state of $SF = (SF0, SF1, \dots, SFN-1)$, that is, $(\forall x/D)F(x)$. Other element than tff (= tN-1N-1) in the matrix has no effect. It means that there is no cross coupling between elements in the set D and an occurrence probability of individual element is independent to each other. If the matrix reaches this state the state transition matrix representation is no more necessary. A conversion matrix between each component is made as shown in Fig. 7. This is the basis of logical expression. It looks too simple but this is the case of the simplest logical representation.

4.4. Learning by Natural Selection

L-matrix of Fig. 6 corresponds to the NN system as is shown in Fig. 8. An NN cannot deal with variable. Some instances are processed. A selector (recognizer) selects instances (a, b, ...z) that are coped with in this system. A sensor (S) of this system accepts information from an external object (EO) (selected instances) to generate a property $F(a)$, etc. A NN (N) accepts it and transforms it to generate output $G(a)$, etc. The output is sent to actuator (or another NN) (A). Assume that a living thing acts according to this

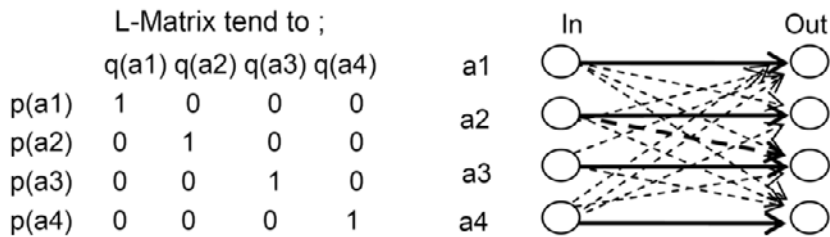


Figure 7. Characteristic of L-matrix.

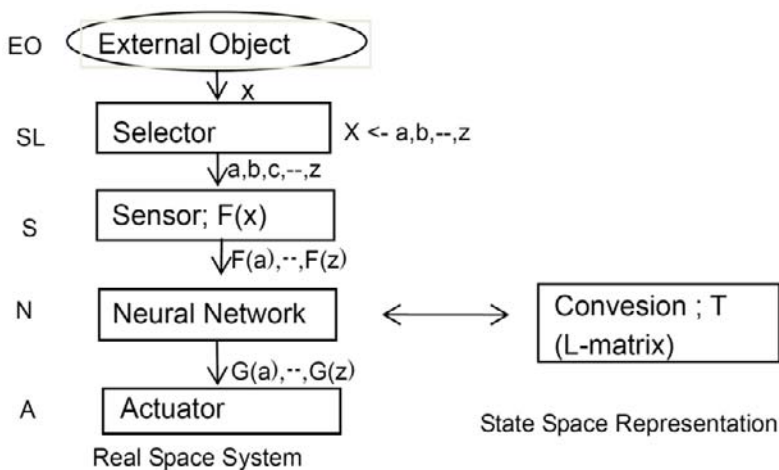


Figure 8. Simplest neural network system.

system. Since a matrix T for the NN is given in a random way when it has born, the result of this movement is not necessarily successful for the purpose, for example, of catching food. The result is fed back and this fact is learned. That is, the weight of paths of NN is modified. To the representation of real conversion corresponds a state space representation. Corresponding to the modification of real conversion matrix the representation of a conversion matrix in a state space (imaginary matrix) is modified in such a way as follows. When the input and the output of the NN is $F(a)$ and $G(a)$ respectively and the result was successful, then SFI, SGJ corresponding to ' $F(a)$;True', ' $G(a)$;True' are increased and the remaining elements are reduced. When the result of the movement was unsuccessful, then these are modified in the inverse way. In the early stage of learning the probability of success is not large but, if this subject has a good fortune, then the probability increases by learning. Afterward it can continue to live. Otherwise it cannot survive.

Let a set of $(F(x), G(x))$ pairs that satisfy the logical relation $(Ax/D)[F(x) \rightarrow G(x)]$ is fixed in advance, the state space transition matrix converges to L-matrix. Note that in learning the change of an instance affects plural elements in T . That is, a pair $\{F(a), G(a)\}$ in data relates all states in SF and SG Including ' $F(a)$;True' and ' $G(a)$;True'. Starting from evenly distributed matrix the matrix converges to L-matrix by learning. Because of the lack of space the result is abbreviated.

In reality achieving learning of this type is unnatural because to prepare the set of data pairs like $(F(a), G(a))$ in advance is not realistic. Instead the result of action might be fed back. Learning was assumed made by this feedback. If the cases of success increase while the creature is still living, then the transition matrix tends to L-matrix. Otherwise the probability of success is low and the creature could not survive. The creature with the matrix that corresponds to the more deterministic expression could survive. It is possible to assume that by this natural selection procedure the same result as learning by a set of pairs $(F(a), G(a))$ was obtained. To this learning process in the state space matrix corresponds the real learning process in the real NN. After the L-matrix is reached in the state space, a corresponding real matrix results in the real space.

This corresponds to a language expression and thus the creature could have the concept of function (behavior). What made it was the data in the nature that exemplify the hidden logical relation. It reveals the fact that what produces L-matrix in a living thing is external information. When the external information meeting the relation $(Ax/D)[F(x) \rightarrow G(x)]$ is used for learning, the NN reduces to L-matrix corresponding to this formula. Information embodying basic characteristic of language existed from the beginning in the nature. Human being did not create but discovered the logical relation and embodies the basis of language.

4.5. L-Matrix for the More General Form of Predicate

L-matrix is decided depending on logical form. What was shown in Fig. 6 is the case of simplest predicate. For the more general cases like $(\forall x/D)[F_1(x) \wedge F_2(x) \rightarrow G(x)]$ (plural predicates) or like $(\forall x/D)(\forall y/E)[F_1(x) \wedge F_2(x, y) \rightarrow G(y)]$ (plural variables), the more complex forms of L-matrix are generated. For example, for the former predicate a compound state $\mathbf{SF} = \mathbf{SF}_1 \times \mathbf{SF}_2$ is defined where \mathbf{SF}_1 and \mathbf{SF}_2 are the states corresponding to $F_1(x)$ and $F_2(x)$ respectively. Correspondingly a compound probability vectors \mathbf{PF} for state \mathbf{SF} are defined. Accordingly number of states in \mathbf{SF} becomes $2^{**}(2^{**}n)$ where $2^{**}n$ denotes 2^n . In case of plural variables being involved in a formula like $(\forall x/D)(\forall y/E)[F_1(x) \wedge F_2(x, y) \rightarrow G(y)]$, a new variable z defined over $D \times E$ is introduced. In these cases three dimensional (cubic) matrix or yet higher dimension matrix appears. The latter case is illustrated in Fig. 9. The first, second and third axes of the state space representation represent $F_1(x)$, $F_2(x, y)$ and $G(y)$ respectively. Every component of this cubic matrix is decided in the same way as was described above for the simplest case.

4.6. Including Ambiguity in Logical Expression

L-matrix had many zero elements. But this was for deterministic logical expression as was shown as $(Ax/d)[F(x) \rightarrow G(x)]$. When some ambiguity is included in logical expression, L-matrix must be different from that. Assume to include a probability measure in logical expression. A simple fact expression like $(\forall x/D) F(x)$ is extended to $(\forall x/D)\{F(x), p(x)\}$. It means that probability of ' $F(x)$; True' is $p(x)$ where $p(x)$ is a probability distribution over D .

$p(x)$ over D and state probability distribution \mathbf{PF} over the state Set \mathbf{SF} are exchangeable. Let's assume that ' $F(x)$; True' for $x; i, j, \dots$, ' $F(x)$; False' for $y; k, l, \dots$ in

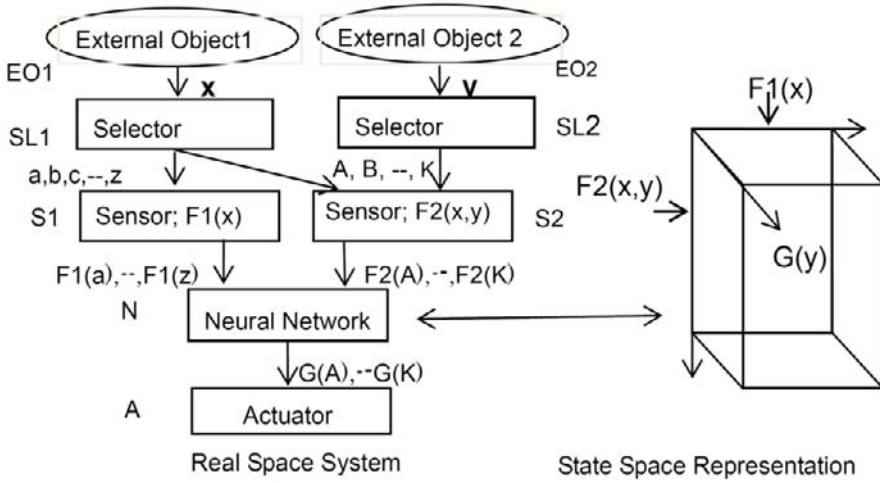


Figure 9. Multi-dimensional representation of transition matrix for $(\forall x/D)[F1(x) \wedge F2(x, y) \rightarrow G(y)]$.

SFI Then $PFI = p(i) \times p(j) \times \dots \times (1 - p(k)) \times (1 - p(l)) \times \dots$ and $p(x) = \sum_{*I \in I} PF_I$ where $*I \in I$ denotes to sum up positive components with respect to i in \mathbf{PF} .

On the other hand probability is introduced in logical implicative formula like $(\forall x/D)\{[F(x) \rightarrow G(x)], q(x)\}$. It means that $F(x) \rightarrow G(x)$ holds with probability $q(x)$. This is only the case the probability measure can be introduced keeping independence of element. Then logical inference is performed in two steps; ordinary logical inference and probability computation.

$$(\forall x/D)\{F(x), p(x)\} \wedge (\forall x/D)\{[F(x) \rightarrow G(x)], q(x)\} \Rightarrow (\forall x/D)\{G(x), r(x)\}$$

$$r(x) = f(p(x), q(x)), \text{ a function of } p(x) \text{ and } q(x).$$

The evaluating of $r(x)$ is, $r(x_i) = \sum_{*i \in I} PGI = \sum_{*i \in I} (\sum_I PFI \times t_{ij})$, (x_i is i -th element of D).

In L-matrix for this case, many elements are no more zero but have certain positive values because $(\forall x/D) [F(x) \rightarrow G(x)]$ is assured only with probability $q(x)$. It is an extension of L-matrix. While L-matrix for non-probabilistic case was specified uniquely to a logical expression, there is a class of matrix representations which are different to each other dependent on the probability $p(x)$ and $q(x)$ for the probabilistic case. It is not an arbitrary matrix but is restricted to the class that meets the relation between \mathbf{PF} and \mathbf{PG} obtained from $p(x)$ and $r(x)$ respectively as above.

4.7. NNs with L-matrix as Language Processor

NNs with L-matrix in the state space behaves like a language processor because it accepts $F(x)$ and Generates $G(x)$ according to the logical relation. Even in the probabilistic case the learning brings an arbitrary NN to the one that is equivalent-to-language. It is also a way to memorize the logical expression in the form of NN. To refer this NN is equivalent to use knowledge. This NN can be a concept of function (behavior) that existed before language.

In general, NN has arbitrary transition matrix (M) and the difference between processing by NN and language processing is difference of arbitrary matrix (M) and L-matrix. Range of representation of general NN is wider than that of language because M include L. Some information such as to represent emotion, feeling, skill and so on are in the class M-L. It can not be represented in language.

5. Making Model of Language Origination

Language was made at some time in the history of human being. Today we do not have any evidence to show the way it is originated. In the following therefore a model of language origination is made. The model must be made only by means of realizable ways so that it can be simulated sometime in near future. It is to make a model only with NNs with growing and learning capability.

5.1. Primitive Language

Various concepts are processed in a human. These are classified into two classes; (1) primitive concept and (2) compound Concept.

[1] Primitive concept represented by physical object

Primitive concept is the one that cannot be decomposed further and dealt directly with by a single NN. There are various types of primitive concepts. These are either (1) entity or (2) property of entity or (3) relation of entities or (4) function (behavior) of entity(s). These are what relate closely to human life or activity. An entity is detected by a sensor, a property of an entity is designated by a specific sensor that detected the entity (the sensor be an identifier of this property), a relation of entities is detected as a co-appearance of entities and is designated by the sensors that detected the entities. The concept of function (behavior) is represented by L-type NN.

In any case an existence of neuron represents a concept. A concept becomes a real concept by being referred. The reference is made by connecting mutually NNs. If an action NN is connected to a voice generator, then voice corresponding to this concept (function/behavior) is generated.

[2] Compound concept realized by NNs

Compound concept is the one constructed from existing concepts. By connected primitive actions a compound action is formed. Compounding enables one to achieve a more complex action. For example; a primitive food capturing action is only to capture food by chance while a compound food capturing action enables one to capture food more efficiently, for example, by sensing, approaching and capturing object.

Compounding can be achieved either by serial or parallel connections or their mixture of NNs. In this case if every NN in a serial connection is of L-type, then compound processor is also of L-type because L-matrix multiplied by L-matrix reduce to L-matrix. In a compound processor any serial paths is of L-type. Serially connected NN is equivalent to NN cascade [11] and it is known that to design a NN cascade in order to achieve a complex objective than to realize it by a single large NN because the scope of necessary learning become narrow. Cascading L-Type NN is easy.

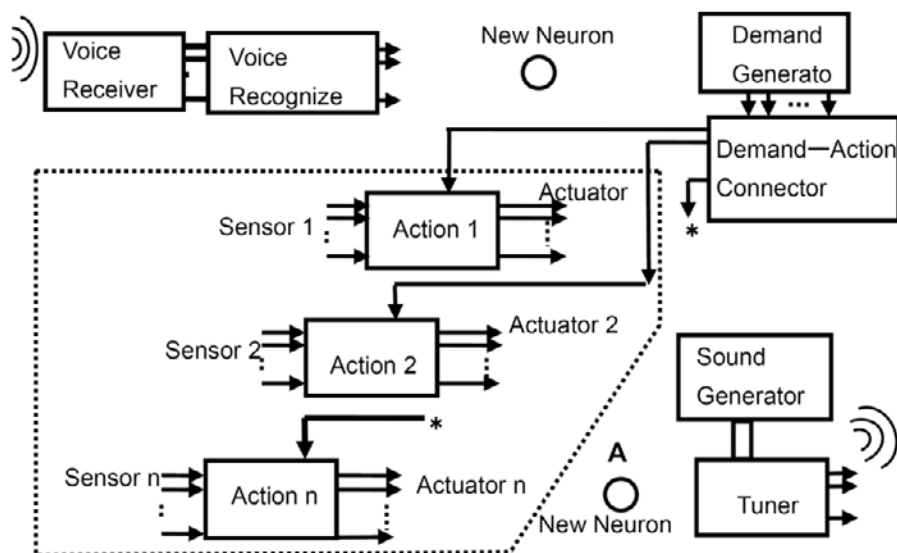


Figure 10. Information Processing Mechanisms Before Language.

5.1.1. Generation of Voice Language

Primitive language is assumed to be a holistic voice language. It is to utter (non-structured) voice corresponding to internal concept and to make actions by hearing voice of the others. Assume that a mechanism for generating sound and tuning had already been made in the process of evolution. Voice tuning is achieved by controlling tongue motion, that is, by controlling muscles around tongue bone. To generate different voices is to send different signal to relating muscles. Uttering is to connect concept to voice generator. Assume also that a mechanism to recognize voice and other signals from the others had been made.

It is shown in Fig. 10 that recognizer, actions and voice generator exist separately. Action is fired by demand such as food capturing requirement evoked by hungry sensor. There are many different demands. It shows an inner state of living things before language was acquired.

Assume a new neuron has been born and grew as is shown by A in the Figs 10 and 11.

Assume also that it connected action and voice generator. Then a voice was generated corresponding to action. The voice was determined by the state of the NN at the time and there was no certain rule to decide it but it was decided in the random way.

5.1.2. From Personal to Social Language

Primitive language generation model mentioned above is of personal language. No one can understand the uttered voice. For the purpose of communication any language must be social language. It means that concept-voice relation must be made common to all members in the same community.

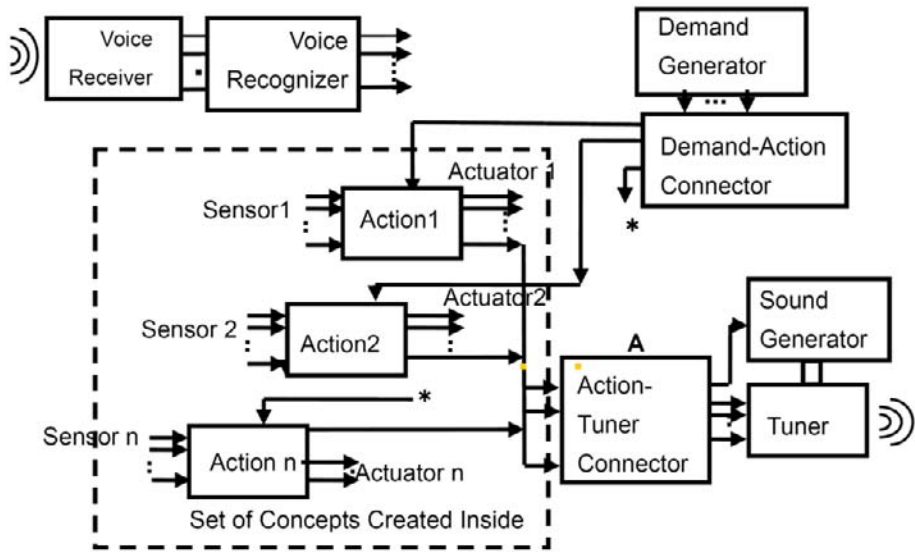


Figure 11. Action is connected to voice generator.

Here is S. Kirby’s interesting simulation [12]. To every person a concept-voice relation is given randomly. No one could not understand to each other. But after frequent contacts among members, a stable voice-concept relation was made in the community. It can be explained as that learning was made to make the conversion part of this new NN the same as the others.

5.2. Toward Compositional Language

The above model of originating primitive language does not include the concept of the other person. The subject of action was always the self. Therefore it is a subjective model. It was not necessary to represent explicitly the subject of action explicitly and the language could be holistic one. If the subject of the action can be the other person than him-/herself, then an objective representation at least of (Subject-Action) form is necessary. If this action has an object, then (Subject-Action-Object) form is necessary.

5.2.1. Identification of the Others – Mirror Neuron

How can the concept of others obtained (as subject of action)? This problem is closely relating to mirror neuron [13] that was discovered in middle 1990 in the monkey brain first, and afterward in the human brain by making use of fMRI. This neuron is activated when the subject imitates others. Its detail however is not yet made clear.

Here is a question on mirror neuron. Does mirror neuron have special structure different from ordinary one or is only its function special? At the moment, there is no clear analysis of mirror neuron. However it is possible to say like this. If it is of special structure, why was it not found earlier? It is also said that imitation can be realized by compound of ordinary neurons.

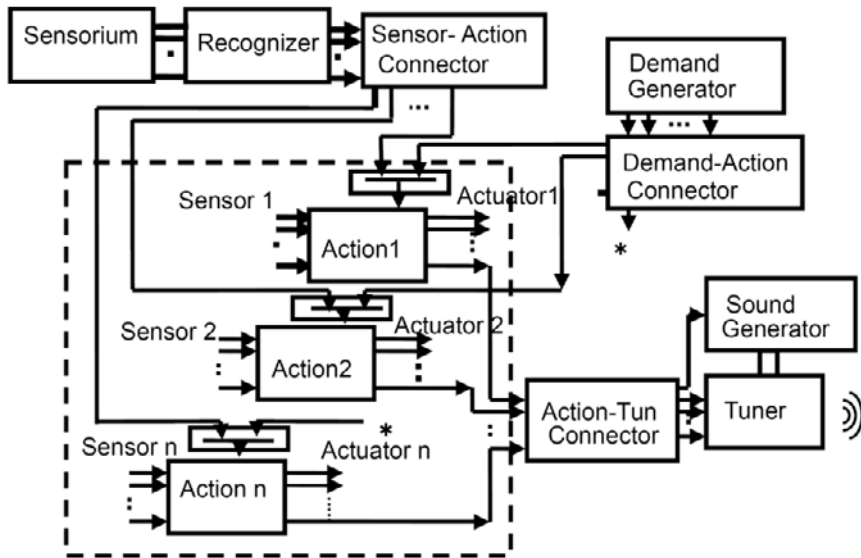


Figure 12. Model of action imitation.

Imitation of action is represented as follows. If {Subject(other)-Action(of other)-Object(other's action's)} is recognized then {Subject(self)-(the same) Action(of self)-Object(self action's)} is activated (Fig. 12). That is, the feature of imitation is,

- (1) Activated by output of recognizer.
- (2) Output of imitating neuron activates self action(s).

To enable activation of action by the external signal is to connect output of recognizer to an internal action. It may be represented like the one as shown in Fig. 11.

It is possible to explain some evidences by this model. It is found that Wernicck Area.

In the left temporal lobe of brain understands voice language and Broca Area in the frontal lobe of brain generates voice language. Wernicck area is connected to Broca area. It was observed that,

- (a) A patient having defect in Broca area could understand voice language but could not speak it. It can be interpreted as action-tuner connector is in Broca area and destroyed.
- (b) A patient having defect in Wernicck area could speak voice language but could not understand it. It can be explained like that voice-action connector is in Wernicck area and destroyed.

5.2.2. Imitation Model

Imitation is realized by making a model as follows. Recognizer is composed of action recognizer and object recognizer. Self action is activated by the object recognizer. It means that an action of recognized object is imitated. It is thought that some animal's behavior to follow its mother can be explained by this model. If the voice tuner connected to the action is activated, then its voice expression is made.

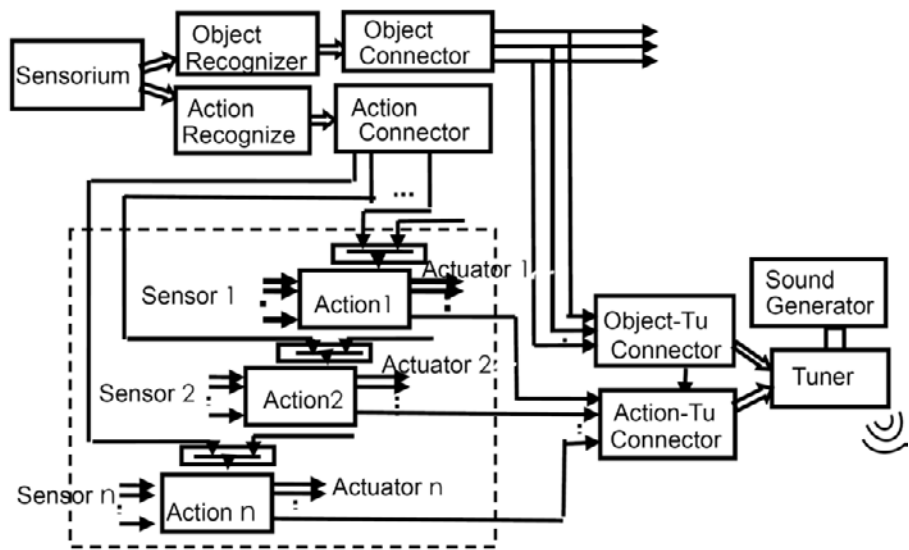


Figure 13. Model to imitate specific object.

5.2.3. Compositional Language by Sequencing Control

The fact that the other person is recognized and is uttered means to articulate language. It is the start of structuring words toward compositional language. Sometimes it was argued that the compositional language was made first. But it is more natural to think that a structure of concepts was made first, then the corresponding structure of language (Fig. 13).

In order to make compositional voice language, a control of order of uttering voice is necessary. Sequencing must be made between voice expressions of subject, action and object. It is possible to make a specific way of ordering by activating action-tuner connector by the result of subject-tuner connector. But there is a requirement for wider class of more general sequence controls in living thing. Today a lot of neurons are activated by a unified control by brain through the nerve center.

5.3. Shift to Symbolic Language

It is thought that symbolization started after the compositional voice language was used. For symbolization primitive concept must be symbolized, memorized and referred by the symbol. It is to make a world of symbols that is completely different from but corresponds to a real world. But how was it done? What was the first symbol? How is the symbol memorized? Many things are still in the dark.

Invention of letter might have a large effect on symbolization. Here is an observation related to this matter. A patient had a defect in part of language understanding. It was observed that when this patient read letters the read information was transmitted to angular-gyrus via primary visual cortex. The read letter was matched with what was uttered and form of voice expression was processed in Wernicke area. In the brain of normal person reading letter, Wernicke area nor angular-gyrus did not activated. This

observation tells us that visual information is directly processed and understood today. But this part of brain must be made recently because the invention of the letter can go back at most 10,000 years ago. Voice understanding is done by old part of brain while visual information path exists already independently for processing symbol.

Symbolization brought a greater progress in language and intelligence. It enabled one to represent multi-layered concept. It enables one to represent a very complex situation (and knowledge) in language. Multi-layered representation requires to let a language expression as an object of description by language. In case of NN, it is to make the other NN an object of an NN processing. But it is difficult. After the world of symbols has been established everything has been done in the world. Rule of structuring language was made and various language structures were made symbolically. In this way new (hypothetical) concept could be created by symbolic operation. It became a new real concept after proved. It seems sure that invention of letter accelerated it.

If L-type NN is identified and can be referred by a symbol, it could be a prior-stage of symbolization. But the mechanism of symbolization is still unknown. It is remained as a future problem.

6. Some Related Issues to Origin of Language

There are some important issues related to the origination of language such as the questions like ‘why could only human being have language?’ and ‘why could human being make many voices?’, etc. But because these issues do not directly relate our objective, the further discussions are abbreviated.

7. Conclusion

This paper discussed the origin of language after mentioning the relation between intelligence and language. Research areas such as; anthropology, biology, brain science, information science, neuro-science and so on relate this issue. Fragmental pieces of evidences and discoveries from these areas were used and a thin thread leading to goal was made. A model of language generation based on artificial NN was made by connecting existing knowledge and bridging the remained gaps by conjecture.

The main issue was the relation between biological processing and language. These are different but under the certain condition they can be the same. NNs that meet this condition can represent and store knowledge. It connects very realistic processing by neurological element to language. It can be concluded by the analysis of biological and language processing that logical inference is the basis of meaning of language processing and language was not created by human being but existed from the beginning like some physical/mathematical truth existed from the beginning. What could human being do was to discover it.

A model of generating language based on artificial NN was made. Whether this model is acceptable or not must be examined by executing simulation actually.

This is the very entrance of symbolic language processing. Nothing could be found on the way of symbolization. Including this issue the area of matters related to intelligence and language is vast. Lot of problems remain unsolved and are left to future re-

search. Among all to discover a way purely symbolic language was born in human brain is the biggest one. As well to know the effect of the invention of letters brought into language is also vital importance.

This research was made in the top-down way. Different from the ordinary scientific (bottom-up) approach to accumulate evidences, top-down approach can show only a possibility. Since bottom-up approach need too long time to goal and often loose way to go, discussion of the possibility is very often necessary.

Reference

- [1] Y. Ohsawa; Data Crystallization: Chance Discovery with Unobservable Events, *New Mathematics and Natural Computioin*, Vol. 1, No. 3, World Scientific, 2005, 373-392.
- [2] *Handbook of Data Mining and Knowledge Discovery*, (eds. W. Klösgen and J.M. Żytkow) Oxford Univ. Press, 2002.
- [3] M. Tallerman (ed.); *Language Origins: Perspectives on Evolution*, Oxford University Press, 2005 [3-1]. S. Mithen; *The Singing Neanderthals-The Origin of Music, Language, Mind and Body* Weidenfeld & Nicolson Ltd, London, 2005.
- [4] S. Ohsuga; Intelligence for upgrading information, in *Web Intelligence Meets Brain Informatics*, Springer LNAI 4845, 2006.
- [5] K. Nakamura; A thermosensory pathway that controls body temperature, *Nature Neuroscience* 11, Jan. 2008.
- [6] J. Vaario; *An Emergent Modelling Method for Artificial NNs*, Doctor Thesis, Univ. of Tokyo, 1993.
- [7] P. Prusinkiewicz and A. Lindenmayer; *The Algorithmic Beauty of Plants*, Springer-Verlag, 1990.
- [8] A. Wray; *Protolanguage as a Holistic Systems for Social Interaction*, *Language and Communication*, 18, 47-67, 1998.
- [9] W.H. Calvin and D. Bickerton; *Lingua ex Machine: Reconciling Darwin and Chomsky with the Human Brain*, Cambridge, MA: MIT Press, 2000.
- [10] S. Ohsuga; Symbol Processing by Non-Symbol Processor, *Proc. PRICAI'96*.
- [11] S. Fahlman and C. Lebiere; *The Cascade-Correlation Learning Architecture*, created for National Science Foundation, Contract Number EET-8716324.
- [12] S. Kirby and M.H. Christiansen; From Language Learning to Language Evolution, in *Language Evolution*, (eds. M. H. Christiansen and S.Kirby), 272-294, Oxford University Press, 2003.
- [13] G. Rizzolatti, L. Craighero; The Mirror Neuron system. *Ann.Rev. Neuroscience*, 27, 169-192, 2004.

Multi-Agent Knowledge Modelling

Marie DUŽÍ,^a Anneli HEIMBÜRGER,^b Takehiro TOKUDA,^c Peter VOJTÁŠ,^d and
Naofumi YOSHIDA^e

^aVSB-Technical University Ostrava, Czech Republic; marie.duzi@vsb.cz

^bUniversity of Jyväskylä, Finland; anneli.heimburger@jyu.fi

^cTokyo Institute of Technology, Japan; tokuda@cs.titech.ac.jp

^dCharles University Prague, Czech Republic; peter.vojtas@mff.cuni.cz

^eKomazawa University, Japan; naofumi@komazawa-u.ac.jp

Abstract. This paper contains five contributions of the participants of the panel discussion on “Multi-agent Knowledge Modelling” in the EJC 2008 conference. We addressed four main topics: (a) Semantic Web technologies, (b) reality vs. agents, (c) cross-cultural knowledge and (d) communication of agents. Each of the discussants presented his/her view of the addressed problem. Some views were rather pessimistic, other optimistic or realistic, but all of them posed questions and raised open problems as well as solution proposals. Thus this paper is a contribution to the topic of multi-agent knowledge modelling from different points of view.

Keywords. Knowledge modelling, Semantic Web, annotation, autonomous agents, ontology, logical analysis of natural language, communication, cross-cultural knowledge.

1. Introduction

The technology of multi-agent systems poses new challenges in the area of distributed knowledge modelling and management. This paper opens a discussion on the topic of multi-agent knowledge modelling from different points of view. The paper is based on the contributions presented in the panel discussion of EJC 2008 Conference.

The Semantic Web is a new slogan that became almost fashionable in the web environment of the multi-agent world. However, there is not much agreement on the meaning of this slogan, as well as on the technologies which should make it possible to meet the proclaimed goals of the new web generation. Different views to this issue are discussed in Section 2. Here we address the problems connected with ontologies, semantic annotation and knowledge interface between real and virtual multi-agent world. In the last paragraph of this section we discuss relationship between culture and knowledge in multi-agent systems. Particular human/computer agents have to communicate with each other in order to achieve their individual as well as collective goals. In Section 3 we deal with the problems connected with communication, knowledge representation and ontology from a logical point of view. The TIL language of constructions is presented here. This language is suitable for a fine-grained logical analysis of particular messages, which in turn makes it possible for agents to communicate in a near-to-natural human-like way.

The paper is not a coherent paper presenting the solution of a particular problem. Instead, it is a multiple-viewpoint contribution to one and the same problem of multi-agent knowledge modelling, which does not propose a ready to use recipe. Rather, different open problems are discussed and different provoking proposals presented.

2. Semantic Web technologies

2.1 Pessimistic view of Semantic Web agents and ontologies (by Takehiro Tokuda)

In this section we present a rather pessimistic view of Semantic Web agents. In case of a closed and easily controlled environment the agents can be successfully designed in order to execute their individual goals. However, our pessimism concerns the case of open Web environments. The idea of Semantic Web agents is promising but far from being easily realisable. The vision of Semantic Web agents is closely related to the vision of the Semantic Web itself, which was introduced in Semantic-Web Road Map in 1998. In essence, Semantic Web is a shift from “Web for humans” to “Web for both humans and programs”. Hence it is a realization of machine recognizable or machine processable Web.

The vision of the Semantic Web was described in May 2001 issue of Scientific American magazine (see [7]). In the beginning of this article, Pete and Lucy look for a physical therapy agent and try to make an appointment with the expert. They make use of their ‘computer agents’. Here is a part of the description of the results. “At the doctor’s office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom’s prescribed treatment from the doctor’s agent...”. This is an optimistic vision, isn’t it?

However, from the viewpoint of pessimistic/optimistic approach, we are rather pessimistic concerning the current stage of *Natural Language Processing* and optimistic concerning *Machine Processable Metadata*.

Pessimistic views are often presented by invited speakers in the Semantic Web conferences or panelists in the World Wide Web conferences. The open, frequently addressed problems are these:

1. RDF metadata level is widely applied, but the RDF semantics is very coarse-grained. OWL ontology level is not so widely applied, though the semantics and expressive power of OWL is broader than that of RDF.
2. The emphasis on complete logical inference systems and automated theorem provers seems to be too demanding. The pursuit of powerful and expressive ontology languages is driven by an opposite goal. We need rich and expressive semantics in order to know *what* is there first and only afterwards to infer *some relevant* consequences.
3. The possibility of building up large-scale ontologies is still limited though such ontologies are highly desirable. Small-scale ontologies are currently much more likely to come into being.

There is, however, still an open problem how to understand the notion of *ontology*. Various versions of definitions of ontologies are used in the literature (see [38]). The most frequently used are as follows:

- (a) A collection of terms
- (b) A collection of terms and their various relations (in particular the ‘is-a’ relation)
- (c) A collection of terms with semantic definitions and various relations among them
- (d) A collection of terms, their properties, and their various relations
- (e) A collection of terms, their properties, and machine-processable relations
- (f) A collection of terms, their properties, and machine-processable or non-processable relations
- (g) A collection of terms, or rather concepts used in a given domain, with computer-treatable and executable semantic definitions, as well as the specification of their mutual relations.

Definition (a) is used, for example, by lay people. Definitions (b), (c) and (d) are used by linguists in natural language ontologies such as WordNet and EuroWordNet. Definition (c) is also used by computer scientists in the database and software system design. Definitions (e) and (f) are applied, for example, by those who use OWL-DL and OWL. Definition (g) might be considered to cover the needs of a full-fledged knowledge representation.

Note that ontologies should be the result of a careful *conceptual analysis* (the term ‘conceptualisation’ of a given domain is sometimes used as well). Thus ontology should be a collection of *concepts* rather than terms. Concepts are the *meanings* of terms, and should thus be language independent. In other words, using an ontology of a particular domain specified, for instance, in English, Finnish, Czech or Japan, the behaviour of the system should not change when switching between particular languages. Unfortunately, such an idea is far from real. Experiences with the EuroWordNet ontology show that we do not even have a unique ontology of human body parts within Europe, though the European languages are similar from the conceptual point of view. The problems even graduate when trying to involve other non-European languages. For instance, in the Japanese language ‘*elder brother*’ and ‘*younger brother*’ are primitive concepts, while ‘*brother*’ is a concept compound from ‘*elder brother*’ and ‘*younger brother*’. On the other hand, in English and other European languages ‘*brother*’ is mostly a primitive concept, while ‘*elder brother*’ and ‘*younger brother*’ are concepts derived from ‘*brother*’.

Concluding this section we would like to state that though ontologies are fundamental for building up the Semantic Web. However, in our opinion, we should not insist on their nice mathematical and logical properties. For instance, there are two different opinions on the decidability of the OWL-DL language. Semantic Web people suppose that OWL-DL is decidable. On the other hand, EJC people believe that even OWL-DL is not decidable. Our feasible image of the use of ontologies may be ad-hoc networks or P2P networks of small processable ontologies.

2.2 Semantic Web technologies from a realistic point of view (by Peter Vojtáš)

When preparing the panel session at the EJC 2008 conference, we tried to discuss a realistic point of view at the Semantic-Web technologies (as opposed to an optimistic and pessimistic one). However, the panel discussion revealed that our point of view is rather pessimistic than realistic. As this paper shows, we agree on rather pessimistic evaluation of the current situation. In this section we discuss the possibility of a

realistic scenario of web-information processing by a machine (i.e., by a computer agent or service). Though the gap between the current Web and the Semantic Web is still large, there are some promising approaches an outline of which we are going to present here.

Lee Feigenbaum, Ivan Herman, Tonya Hongsermeier, Eric Neumann and Susie Stephens in their Scientific American 2007 article [19] briefly summarized the Scientific American 2001 article [7] of Tim Berners-Lee, James Hendler and Ora Lassila as follows:

[...(they)...] unveiled a nascent vision of the Semantic Web: a highly interconnected network of data that could be easily accessed and understood by a desktop or handheld machine. They painted a future of intelligent software agents that would [...] answer to a particular question without *our* having to search for information or pore through results

(quoted from [19], italicized by P. Vojtáš).

Lee Feigenbaum and colleagues conclude that “Grand visions rarely progress exactly as planned, but the Semantic Web is indeed emerging and is making online information more useful as ever”. L. Feigenbaum et al. support their claim by the successful application of the Semantic Web technology in the drug discovery, health care and several other applications. These are mainly corporate applications with data semantically annotated by humans.

Ben Adida in [1] uses RDFa in order to make a bridge from the clickable towards Semantic Web. The RDFa language makes use of human-assisted annotations of *newly* created web resources.

Current web pages seem to be without attraction for anybody if these pages are not provided with a proper annotation. Thus it is necessary to annotate all the pages with an interesting and valuable content. Otherwise their information content might be lost as soon as newly emerging semantic-web applications come into being. In what follows we are going to address the problem of semantization of current web content. We will discuss an automated process of annotation by a semantic repository maker, the aim of which is an adjustment of the current web so that it was accessible for machine processing. In this way the information content of the web gradually becomes at least partly visible to the other users (see [6]).

Scenario. Since current search engines do not fully meet the requirements of users, our starting point is a user (human being, an agent).

User 1 wants to buy a notebook. His/her/its searching strategy is driven by the preferences of price, screen, weight and battery design capacity. In order to combine these preferences, an overall combination score is applied (a selection of the best offer from the Pareto front).

User 2 is searching highly secure cars and highly dangerous streets. To this end the user evaluates traffic accident reports on the web.

Proposal. The goal of the semantization project of the group led by Jaroslav Pokorný and Peter Vojtáš in Charles University Prague is *enhancing the ontology-based annotation* of XML documents and RDFa-annotated HTML files by a semantic repository and user profiles (see [33], [9]).

The main idea is to support automated annotation of existing web resources in a semantic repository.

The first step in our model is a *crawler*. Based on experiences with the Egothor crawler ([18]) we experiment with an extension of functionalities of the corresponding repository (see [20], [5]).

The second step consists in *Web-information extraction*. Here we make use of the extraction based on the analysis of structured pages ([32], [16]). Another technique that we apply is the approach to dominantly linguistic pages ([8]) based on techniques of the Prague school of computational linguistic [30]). After training, the results of this extraction can be used for the automatic annotation of resources ([3]). See, e.g., W3C motivation for ruby annotation (Fig. 1):

Month Day Year	しんかんせん ← <i>ruby text</i>
10 31 2002	新幹線 ← <i>ruby base</i>
Expiration Date	shinkansen ← <i>ruby text 2</i>

Fig.1 W3C motivation of annotation [3]

Such a human-usable annotation has a counterpart in a machine readable annotation. Web information extraction and consequent annotation is the most critical phase. Our experiments support a hypothesis that at least stable pages (for instance such pages which do not change more frequently than once per month) can and should be annotated. Still some contribution of a trained human expert is necessary, and the human-computer interaction improves the quality of such a work in a great deal.

The third step consists in the creation of a *semantic repository* ([9], [33]) and its functionalities. Here we have to solve the uncertainty problem connected with the automatic annotation ([39], [16]). Some functionalities of this repository are already clear, e.g. indexing (both vector model based and XML indexing) and querying (SPARQL, key word queries). Some are based on the model of semantic services ([31]) or models of ontology engineering ([32], [35]), and are to be designed in the future.

The last step consists in modelling a typical *user*, in particular his/her preferences ([16]). We have implemented models serving for user preference querying ([15]).

There is a problem, however, which agents and services (semantic services) will operate on and make use of the semantically enriched repository. Yet even a small step from the corporate web and human annotation towards the Semantic Web and automated annotation extends the scope of information obtainable from the web, and thus contributes to the progress of Emerging Semantic Web. In our opinion, this is a realistic view.

On the other hand, there is also a pessimistic prognosis concerning the current stage of fine-grained knowledge modelling, in particular in the area of human reasoning modelling. In our opinion the realisation of an inference machine that would simulate human reasoning remains unattained strategic goal rather than a reality. Experience with the most successful projects (like, for instance, DeepBlue chess playing system, see <http://www.research.ibm.com/deepblue/>) indicate that we have to combine analytic methods with synthetic ones. Sometimes it is useful to apply a “brute force” calculating method instead of a rigorous analytic method. Similar approach combining analytic and synthetic methods can be observed in medicine. Western-like and Eastern (in particular Chinese) medicines are complementary. Synergy of both

approaches in medicine can be fruitful. Similarly in our attempt for web semantization we have to investigate new non-traditional methods of reasoning.

By way of conclusion we state that the Semantic Web still remains a grand vision. The process of Web semantization is an approach that significantly extends the scope of information obtainable from web. Thus in the age of Information Society it is worth to explore the problem extensively and study particular methods in detail. Some parts of our experiment application passed the quality test of conception. It is a matter of future research to extend the experiments to large data and different kinds of users. Our realism applies to the process of stepwise (possibly distributed) semantic enrichment of web pages *via* the automated third party annotation.

2.3 Reality vs. agents (by Naofumi Yoshida)

In this section we discuss the problems connected with the interface between the real and a multi-agent world. In what follows we address three technical issues: knowledge representation of the real world using *sensor data*, representation of space and time by involving *temporal* and *special* data, and finally *collaboration* of agents. In order to propose some solutions, we discuss a time-space-direction algebra for inter-collaboration among agents and hybrid system design for symbolic and numerical computing by on-the-fly processing.

2.3.1 Knowledge representation of real world by sensor data

In the field of sensor networks and sensor databases [34], various sensors are technically available. They are acceleration sensors, thermograph sensors, location sensors (such as GPS), video cameras, microphones, and so like. These sensors enable us to be aware of the real-world context. It is desirable that the real-world situation be directly reflected in the cyber or agent world. In this way we are able to obtain metadata on the situation in which the agents operate (see, e.g. [41]).

2.3.2 A Time-Space-Direction Algebra for inter-collaboration among agents

The agents of a multi-agent system do not occur in a vacuum. They operate in space and time, and their behaviour has to reflect and influence their environment. Thus temporal and spatial features processing is essentially important for the interconnection between the real world and a multi-agent world.

To this end we proposed a time-space-direction algebra (see [42]). It is a mathematical representation and a formal system for describing physical and electronic objects with temporal, spatial, and directional features. In principle, the system makes it possible to describe any behaviour of physical or electronic objects. The temporal and spatial relationships among two objects have been designed (see [2], [17], respectively). By adding direction and set-operators, we define a unified description form as the time-space-direction algebra.

2.3.2.1 Data structure of Time-Space-Direction Algebra

Every single physical and electronic object has the following basic attributes:

- ID (an identifier of an object)
- Point1 (x_1, y_1, z_1 : a spatial position an object)
- Point2 (x_2, y_2, z_2 : a direction of an object is defined by a vector from Point1 to Point2)
- Timestamp (date and time)

The data structure ‘tuple’ for describing a snapshot of an object is defined as follows:

- (ID, $x_1, y_1, z_1, x_2, y_2, z_2, t$)

The data structure ‘sequence’ for describing behaviours of objects with temporal, spatial, and directional features is defined as follows (see Fig. 2):

- (ID, $x_{11}, y_{11}, z_{11}, x_{21}, y_{21}, z_{21}, t_1$),
- (ID, $x_{12}, y_{12}, z_{12}, x_{22}, y_{22}, z_{22}, t_2$),
- ...,
- (ID, $x_{1n}, y_{1n}, z_{1n}, x_{2n}, y_{2n}, z_{2n}, t_n$)

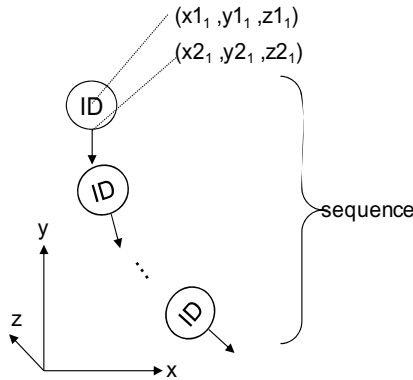


Fig. 2: Data Structures of Time-Space-Direction Algebra

2.3.2.2 Operators of Time-Space-Direction Algebra

Operators of the Time-Space-Direction Algebra are the following ones (see [42]): Temporal Comparison-Bi-Operators, Spatial Comparison-Bi-Operators, Arithmetic-Bi-Operators, Uni-Operators, and Matching-Operators. Fig. 3 shows the temporal and spatial operators (Temporal Comparison-Bi-Operators and Spatial Comparison-Bi-Operators). Fig. 4 shows examples of Arithmetic-Bi-Operators.

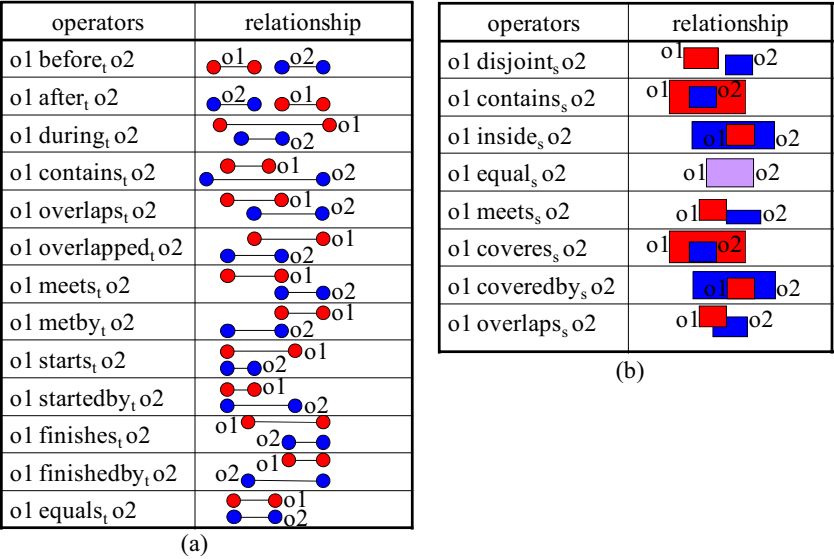


Fig. 3: Temporal (a) and Spatial (b) Operators

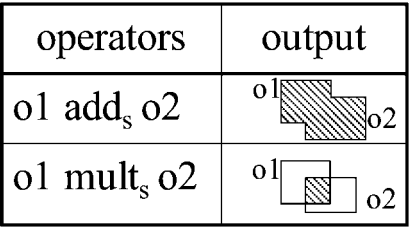


Fig. 4: Examples of Arithmetic-Bi-Operators (Add and Mult (multiply) Operators)

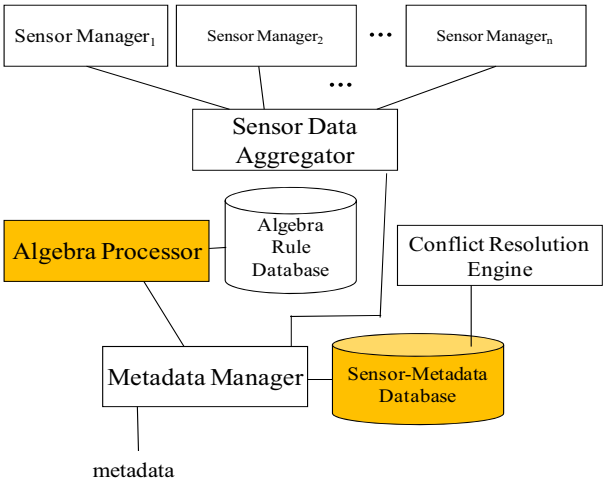


Fig. 5: Applications of Time-Space-Direction Algebra for Metadata Extraction

2.3.2.3 Applications of Time-Space-Direction Algebra for metadata extraction of agents

The Fig. 5 illustrates the real-world metadata extraction. Using the Algebra, we can handle the real-world data extraction. In this way we model the real world and agent's interconnection. Thus the behaviour of real objects is reflected by behaviour of agents in a multi-agent world. Since the Algebra enables us to describe any behaviour of physical or electronic objects including their temporal, spatial, and directional features, we can calculate temporal, spatial and directional relationships. It also enables us to reduce the computational complexity of algebraic expressions.

2.3.3 Hybrid System Design for Symbolic and Numerical Computing by On-The-Fly Processing

When building up a multi-agent system, we have to deal with many conflicting factors and goals, such as symbolic vs. numerical computing, cyber vs. real world, optimistic vs. pessimistic view, and so on. In general, there is a simple solution: a *hybrid system* approach. The application of a 'Meta' or 'Hybrid' system turns out to be simple and effective (see, e.g. [43]). Moreover, the 'On-The-Fly hybrid/meta system' is more effective than the previous versions of on-the-fly or hybrid/meta systems due to the dynamic interconnection between heterogeneous systems. In this way we are able to obtain data on the advanced system environment. Moreover, using the system, an advanced information retrieval is made possible as well (see, e.g., [40] [44]).

2.4 Culture and Knowledge in Multi-Agent Environments (by Anneli Heimbürger)

2.4.1 Culture at Five Levels

Globalization is one of the main trends in our world. Increasingly, eastern and western cultures meet each other in connection with business, governmental and environmental issues, research, education and tourism. Professional like business executives, project managers and project team members are finding themselves in uncertain situations due to culturally dependent differences in communication protocol, language and value systems. Cross-cultural communication is a current topic in many multi-cultural organizations and companies. In cross-cultural world many collaborative actions are carried out in virtual and physical environments such as email, telecons, Web meetings, virtual spaces, face-to-face meetings, workshops and conferences. Some examples of differences between eastern and western cultures that we may face are meeting protocols, formality and rituals, orientation to time, communication style and decision-making process.

When we are talking about the concept *culture*, it is very important to understand its different levels. According to King [26] cultures can be considered at four levels: national cultures, organizational cultures, organizational subcultures and subunit cultures. We extend King's categorization with team cultures. Related to national cultures one of the mostly cited studies is Hofstede's framework for cultural dimensions ([24], [23], [27]). It is based on questionnaire study in 74 countries and on statistical analysis of the survey data. Organizational culture is characterized by consistency across individuals and units in terms of assumptions, values and artifacts. Assumptions are formed over time as members of an organization make decisions, cope with problems and take advantage of opportunities. Values are a set of social

norms. Artifacts are visible aspects of an organizational culture, for example a knowledge repository system. Organizational subcultures may reflect organizational structure, professional occupations, task assignments, rank in hierarchy or technologies used. Subunit cultures are created within the boundaries of particular subunits of an organization. Team cultures are mechanisms for individuals with diverse specialized knowledge to work towards a common goal. Teams are typically focused on a single objective and they are temporary. If all team members are from the same organizations, the team culture reflects the organizational culture. In multi-organizational projects, many team cultures may collide or softly meet depending on the cultural competence of the team members and the ICT systems they are using.

2.4.2 What is Cross-Cultural Knowledge?

Cultural knowledge, cultural awareness, and cultural sensitivity all convey the idea of improving cross-cultural capacity ([24], [25]). Cultural knowledge is familiarization with selected cultural characteristics, history, values, belief systems, and behaviours of the members of another ethnic group. Cultural awareness means developing sensitivity and understanding of another ethnic group. This usually involves internal changes in terms of attitudes and values. Awareness and sensitivity also refer to the qualities of openness and flexibility that people develop in relation to others. Cultural awareness must be supplemented with cultural knowledge. Cultural sensitivity means knowing that cultural differences as well as similarities exist, without assigning values, i.e., better or worse, right or wrong, to those cultural differences. Cultural competence has become one important dimension for the success in today's international business and research arena. Cultural competence is defined as a set of congruent behaviours, attitudes, and policies that come together in a system and/or among professionals and enables the system and/or professionals to work effectively in cross-cultural situations.

Cross-cultural knowledge can be considered in three main levels:

- Level 1. Explicit knowledge, for example temporal facts (holidays, festivals, business hours, academic terms) and geographical facts (cities, climate, people, language etc.)
- Level 2. Reported knowledge based on survey data and/or field studies (meeting protocol, formality and rituals, orientation to time, communication style and decision-making process)
- Level 3. Tacit knowledge, for example organization, project and team specific knowledge. Tacit knowledge is often classified.

In modelling, designing and implementing culture-sensitive information systems and applications more detail contextual analysis is needed. Computer science and operational-oriented definition of the notion *context* is as follows ([4], [10]). Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application.

How do cultures relate to knowledge management? Holden in [25] discussed culture shapes assumptions about which knowledge is important. Culture mediates the relationships between organizational and individual knowledge. Culture creates a context for social interaction. Culture shapes processes for the creation and adoption of new knowledge. Creating, analyzing, delivering, sharing and managing contextual cultural knowledge is a challenge.

2.4.3 A vision of a Cross-Cultural Multi-Agent System

We discuss the question of how could we possibly apply multi-agent technology to support cross-cultural communication and overcome possible barriers between eastern and western cultures. Multi-agent systems (MAS) can be embedded for example into business processes in the company, learning management systems, meeting systems, virtual collaborative working environments, and email systems ([21], [22]). The agents perceive their environment and initiate certain actions in their environment. They are able to recognize the state of particular tasks and perform operations needed to meet the goal of a task. Based on reasoning the agents can execute actions to facilitate the performance of the task. For example, a Japanese-Czech-Finnish multi-agent system could serve as a guide to formulate appropriate emails to counterparts in Japan and in the Czech Republic. In this example, MAS serves as a platform in which the cross-cultural communication can be realized with the support of culturally aware agents that can help people to understand better their counterparts in interaction.

Let's study the following example of decision making situation as a visionary example how a multi-agent system could behave and support a cross-cultural team. A team consisting of a Czech, Finnish and Japanese people are discussing via email about scheduling certain project tasks. A Czech team member raises a question to the Japanese team member: "Should we execute the tasks A and B in a successive way, i.e. one after the other is finished, or should the tasks be performed in parallel, say in the overlapping two weeks?" The Japanese team member answers simply "yes". The Czech and Finnish members are a bit confused and repeat the question again. They keep getting the same positive answer. Meanwhile the Japanese team member is totally happy with the situation. The team work is in a deadlock.

Now the support agents come into the scene. The Czech agent is a logician, and thus he/she/it deduces: "In Japan, different views can co-exist in space and time, without invoking a conflict or a need to resolve the differences. Thus the question has been understood as expressing an 'inclusive or'. In Western logic, contradictory views of the same thing are not tolerated. Differences must be resolved. Singular and exclusive truths are looked for". The Finnish agent advises that the question should be formulated in a different way. The Japanese agent is listening to the communication between the Czech and Finnish agents, and it also stores the discussion content into its knowledge base. (This is an example of agents' learning.) The Japanese agent compares this knowledge with its previous experiences and infers that "the Japanese people try to avoid saying 'no', or 'I don't know', because this is considered impolite"; the agent informs the others about this conclusion. The Czech and Finnish team members are still a bit confused. They ask their agents what we shall do now. However, the Czech and Finnish agents learned the conclusion of the Japanese agent, and their advice is double-dealing: "Agree for now; and next formulate some separate questions". The first part is inferred from their knowledge on "harmony and politeness is a very important issue in Japanese culture". The second part is inferred from their knowledge on the need to resolve the situation. Thus the Czech and Finnish team-members conclude: ok, we have to reformulate the question, perhaps as follows:

- Which are the common subtasks of A and B?
- What is the deadline for the task A?
- What is the deadline for the task B?

If no answer satisfactory in Western standards is obtained, then they have to decide on their own. After all, harmony and peace are the most important aspects of a team work.

Organizations that will use such advanced cross-cultural human-computer systems should also provide a source of learning experiences and innovative thinking for their personnel in order to enhance the competitive position of the organizations.

3. Communication of agents

W3C standard languages like RDF and OWL have many limitations with respect to flexible interaction and interoperability among heterogeneous services. We need to develop techniques for autonomous and goal-directed agent interactions, agent communication languages that support extended conversations, agent negotiation and methods for peer to peer reactive and proactive agent behaviours in dynamic environments.

According to the FIPA standards¹, *message* is the basic unit of communication. It can be of an arbitrary form but it is supposed to have a structure containing several attributes. Message semantic *content* is one of these attributes, the other being for instance ‘Performatives’, like ‘Query’, ‘Inform’, ‘Request’ or ‘Reply’. The content can be encoded in any suitable language. The standards like FIPA SL and KIF are mostly based on the First-Order Logic (FOL) paradigm, enriched with higher-order constructs wherever needed.² The enrichments extending FOL are well defined syntactically, while their semantics is often rather sketchy, which may lead to communication inconsistencies. Moreover, the bottom-up development from FOL to more complicated cases yields the versions that do not fully meet the needs of the MAS communication. In particular, agents’ attitudes and anaphora processing create a problem (see [12]).

In this section we present a proposal of a rich language apt for agent communication in a near-to-natural human-like way. Agents communicate by messages the patterns of which go beyond the request-response style. In particular, we will to articulate the importance of a formally specified, unambiguous semantics of the language.

3.1 Communication using the TIL-Script language (by Marie Duží)

We need a language that is expressive enough to render all the semantically salient features of natural languages. To this end we use a powerful logical system of Transparent Intensional Logic (TIL), see, e.g., [36] and [37], namely its computational variant, the *TIL-Script* language. The TIL logic system has been described, e.g., in [13] or [14]. The first paper deals with bottom-up vs. top-down approach to Semantic Web technologies. The second paper introduces our new procedural theory of concepts and ontologies based on TIL (see also [11]). TIL language of constructions is an objectual version of a partial, typed, hyper-intensional λ -calculus. ‘Partial’, because we work with partial functions. ‘Typed’, because all the entities the system works with receive types. And ‘hyper-intensional’, because we work within the ramified theory of types,

¹ The Foundation for Intelligent Physical Agents, <http://www.fipa.org>

² For details on FIPA SL, see <http://www.fipa.org/specs/fipa00008/>; for KIF, Knowledge Interchange Format, see <http://www-ksl.stanford.edu/knowledge-sharing/kif/>

which makes it possible not only use the TIL constructions, but also mention them as full-fledged objects. This feature makes it possible to adequately analyse not only simple sentences, but also compound ones, such as sentences with anaphoric references and attitudinal reports.

Transparent Intensional Logic (TIL) is a system with *procedural semantics* primarily designed for the logical analysis of natural language. Traditional non-procedural theories of formal semantics are less or more powerful logical languages, from the extensional languages based on FOL approach, through some hybrid systems up to intensional (modal or epistemic) logics. Particular systems are suited well to analysing restricted sublanguages, and they are broadly used and well standardised. However, the logic of attitudes is a stumbling block for all of them. Moreover, if such a great variety of specialised languages were involved in order to design a communication language for a multi-agent system (MAS), the agents would have to keep switching from one logical language to another, which is certainly not a plausible solution.

On the other hand, TIL, due to its strong typing and procedural semantics, operates smoothly with the three levels of granularity: the extensional level of truth-functional connectives, the intensional level of modalities and finally the hyperintensional level of attitudes. The sense of a sentence is an algorithmically structured *construction* of a proposition denoted by the sentence. The denoted proposition is a flat mapping with the domain of possible worlds. Our motive for working ‘top-down’ has to do with anti-contextualism: any given unambiguous term or expression (even one involving indexicals or anaphoric pronouns) expresses the same construction as its sense (meaning) in whatever sort of context the term or expression is embedded within. And the meaning of an expression determines the respective denoted entity (if any), but not vice versa.

TIL *constructions* are uniquely assigned to expressions as their *algorithmically structured meanings*. Intuitively, construction is a procedure (a generalised algorithm), that consists of particular sub-instructions (constituents). It is an instruction on how to proceed in order to obtain the output entity given some input entities. Atomic constructions (*Variables* and *Trivializations*) do not contain any other constituent but itself; they supply objects (of any type) on which compound constructions operate. *Variables* x, y, p, q, \dots , construct objects dependently on a valuation; they v -construct. *Trivialisation* of an object X (of any type, even a construction), in symbols 0X , constructs simply X without the mediation of any other construction. *Compound constructions*, which consist of other constituents, are *Composition* and *Closure*. *Composition* $[F A_1 \dots A_n]$ is the instruction to apply a function f (v -constructed by F) to an argument A (v -constructed by $A_1 \dots A_n$).³ Thus it v -constructs the value of f at A , if the function f is defined at A , otherwise the Composition is v -improper, i.e., it does not v -construct anything. *Closure* $[\lambda x_1 \dots x_n X]$ is the instruction to v -construct a function by abstracting over variables x_1, \dots, x_n in the ordinary manner of λ -calculus. Finally, higher-order constructions can be used twice over as constituents of composed constructions. This is achieved by a fifth construction called *Double Execution*, 2X , that behaves as follows: If X v -constructs a construction X' , and X' v -constructs an entity Y , then 2X v -constructs Y ; otherwise 2X is v -improper.

TIL constructions, as well as the entities they construct, all receive a type. The formal ontology of TIL is bi-dimensional; one dimension is made up of constructions,

³ We treat functions as mappings, i.e., set-theoretical objects, unlike the *constructions* of functions.

the other dimension encompasses non-constructions. On the ground level of the type-hierarchy, there are non-constructional entities unstructured from the algorithmic point of view belonging to a *type of order 1*. Given a so-called *epistemic (or 'objectual')* base of *atomic types* (\mathbf{o} -truth values, \mathbf{i} -individuals, $\mathbf{\tau}$ -time moments / real numbers, $\mathbf{\omega}$ -possible worlds), the induction rule for forming functions is applied: where $\alpha, \beta_1, \dots, \beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times \dots \times \beta_n$ to α , denoted $(\alpha \beta_1 \dots \beta_n)$, is a type of order 1 as well.⁴ Constructions that construct entities of order 1 are *constructions of order 1*. They belong to a *type of order 2*, denoted by \ast_1 . This type \ast_1 together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, type $(\alpha \beta_1 \dots \beta_n)$, involving \ast_1 in their domain or range is a *type of order 2*. Constructions belonging to a type \ast_2 that identify entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3*. And so on *ad infinitum*.

(α) -*intensions* are members of a type $(\alpha\omega)$, i.e., functions from possible worlds to the arbitrary type α . (α) -*extensions* are members of the type α , where α is not equal to $(\beta\omega)$ for any β , i.e., extensions are not functions from possible worlds. Intensions are frequently functions of a type $((\alpha\tau)\omega)$, i.e., functions from possible worlds to *chronologies* of the type α (in symbols: $\alpha_{\tau\omega}$), where a chronology is a function of type $(\alpha\tau)$. We use variables w, w_1, \dots as v -constructing elements of type ω (possible worlds), and t, t_1, \dots as v -constructing elements of type τ (times). If $C \rightarrow \alpha_{\tau\omega}$ v -constructs an α -intension, the frequently used Composition of the form $[[Cw]t]$, the intensional descent of the α -intension, is abbreviated as C_w . Some important kinds of intensions are:

Propositions, type $\mathbf{o}_{\tau\omega}$. They are denoted by empirical (declarative) sentences.

Properties of members of a type α , or simply *α -properties*, type $(\mathbf{o}\alpha)_{\tau\omega}$. General terms (some substantives, intransitive verbs) denote properties, mostly of individuals.

Relations-in-intension, type $(\mathbf{o}\beta_1 \dots \beta_m)_{\tau\omega}$. For example transitive empirical verbs, also attitudinal verbs denote these relations.

α -roles, offices, type $\alpha_{\tau\omega}$, where $\alpha \neq (\mathbf{o}\beta)$. Frequently $\mathbf{i}_{\tau\omega}$. Often denoted by concatenation of a superlative and a noun ("the highest mountain").

Due to the limited scope of this contribution, we now only demonstrate our vision of autonomous agents by way of example of a simple dialog between three agents, Adam, Berta and Cecil. The agents communicate in a natural, free style way, using anaphoric references, expressing their attitudes and knowledge. Due to the strong typing, we can resolve some ambiguities. The following dialog illustrates the *dynamic method of discourse representation*. For each type, there is a list of discourse variables which are gradually updated in an imperative way by the construction (of the respective type) that received the last mention in the dialog. In our case we have the following discourse variables:

- ind*: individuals,
- loc*: location (GIS coordinates),
- pred*: properties of individuals,
- prof*: properties of individuals (propositional functions),
- rel₁*: relation-in-intension between an individual and a property of individuals,
- prop*: propositions,
- constr*: constructions.

⁴ TIL is an open-ended system. The above epistemic base $\{\mathbf{o}, \mathbf{i}, \mathbf{\tau}, \mathbf{\omega}\}$ was chosen, because it is apt for natural-language analysis, but the choice of base depends on the area to be analysed.

Adam to Cecil: “Berta is coming. **She** is looking for a parking place”.

‘Inform’ message content (first sentence):

$\lambda w \lambda t \text{ } [[^0 \text{Coming}_{wt} \text{ } ^0 \text{Berta}]]$;

(Relevant) discourse variables updates:

$ind := ^0 \text{Berta}$; $pred := ^0 \text{Coming}$;

$prop := \lambda w \lambda t \text{ } [[^0 \text{Coming}_{wt} \text{ } ^0 \text{Berta}]]$;

‘Inform’ message content (second sentence):

$\lambda w \lambda t \text{ } [^2 [^0 \text{Sub } ind \text{ } ^0 she \text{ } [^0 \text{Looking_for}_{wt} she \text{ } ^0 \text{Parking}]]] \Rightarrow$ (is transformed into)

$\lambda w \lambda t \text{ } [^0 \text{Looking_for}_{wt} \text{ } ^0 \text{Berta} \text{ } ^0 \text{Parking}]$.

(Relevant) discourse variables updates:

$rel_1 := ^0 \text{Looking_for}$; $pred := ^0 \text{Parking}$;

$prop := \lambda w \lambda t \text{ } [^0 \text{Looking_for}_{wt} \text{ } ^0 \text{Berta} \text{ } ^0 \text{Parking}]$;

$prof := \lambda w \lambda t \text{ } \lambda x \text{ } [^0 \text{Looking_for}_{wt} x \text{ } ^0 \text{Parking}]$; (‘propositional function’)

Cecil to Adam: “**So** am I.”

‘Inform’ message content:

$\lambda w \lambda t \text{ } [^2 [^0 \text{Sub } prof \text{ } so \text{ } [so_{wt} \text{ } ^0 \text{Cecil}]]] \Rightarrow \lambda w \lambda t \text{ } [^0 \text{Looking_for}_{wt} \text{ } ^0 \text{Cecil} \text{ } ^0 \text{Parking}]$

(Relevant) discourse variables updates:

$ind := ^0 \text{Cecil}$; $rel_1 := ^0 \text{Looking_for}$; $pred := ^0 \text{Parking}$;

Adam to both: “There is a vacant car park at p_I ”.

‘Inform’ message content:

$\lambda w \lambda t \text{ } \exists x \text{ } [[^0 \text{Vacant} \text{ } ^0 \text{Car_Park}]_{wt} x] \wedge [^0 \text{At}_{wt} x \text{ } ^0 p_I]]$

(Relevant) discourse variables updates:

$loc := ^0 p_I$; $pred := [^0 \text{Vacant} \text{ } ^0 \text{Car_Park}]$;

$prop := \lambda w \lambda t \text{ } [\exists x \text{ } [[^0 \text{Vacant} \text{ } ^0 \text{Car_Park}]_{wt} x] \wedge [^0 \text{At}_{wt} x \text{ } ^0 p_I]]$

Berta to Adam: “What do you mean by vacant car park?”

‘Query’ message content:

$\lambda w \lambda t \text{ } [^0 \text{Refine}_{wt} \text{ } [^0 \text{Vacant} \text{ } ^0 \text{Car_Park}]]$

(Relevant) discourse variables updates:

$constr := [^0 \text{Vacant} \text{ } ^0 \text{Car_Park}]$

Adam to Berta: “Vacant car park is a parking lot some places of which are not occupied”.

‘Reply’ message content:

$[^0 \text{Vacant} \text{ } ^0 \text{Car_Park}] =$

$[\lambda w \lambda t \text{ } \lambda x \text{ } [[^0 \text{Parking_lot}_{wt} x] \wedge \exists y \text{ } [[^0 \text{Place_of}_{wt} y x] \wedge \neg [^0 \text{Occupied}_{wt} y]]]]$

Cecil to Adam: “I don’t believe **it**. I have just been **there**”.

‘Inform’ message content (first sentence):

$\lambda w \lambda t \text{ } [^2 [^0 \text{Sub } prop \text{ } ^0 it \text{ } [^0 \neg [^0 \text{Believe}_{wt} \text{ } ^0 \text{Cecil } it]]]] \Rightarrow$

$\lambda w \lambda t \text{ } \neg [^0 \text{Believe}_{wt} \text{ } ^0 \text{Cecil} \text{ } [\lambda w \lambda t \text{ } [\exists x \text{ } [[^0 \text{Vacant} \text{ } ^0 \text{Car_Park}]_{wt} x] \wedge [^0 \text{At}_{wt} x \text{ } ^0 p_I]]]]$,

(Relevant) discourse variables updates: $ind := ^0 \text{Berta}$; ...

‘Inform’ message content (second sentence):

$\lambda w \lambda t \text{ } \exists t' \text{ } [[t' \leq t] \wedge [^2 [^0 \text{Sub } loc \text{ } ^0 there \text{ } [^0 \text{Been_at}_{wt} \text{ } ^0 \text{Cecil } there]]]] \Rightarrow$

$\lambda w \lambda t \text{ } \exists t' \text{ } [[t' \leq t] \wedge [^0 \text{Been_at}_{wt} \text{ } ^0 \text{Cecil} \text{ } ^0 p_I]]$.

And so on.

The role of Trivialisation and empirical parameters $w \rightarrow \omega$, $t \rightarrow \tau$ in the communication between agents can be elucidated as follows. Each agent has to be

equipped with a basic ontology, namely the set of primitive concepts she is informed about. Thus the upper index c_0 serves as a marker of the primitive concept that the agents should have in their ontology. If they do not, they have to learn them by asking the others. The lower index $_{wt}$ can be understood as an instruction to execute an *empirical inquiry (search)* in order to obtain the actual current value of an intension, for instance by searching agent's database or by asking the other agents, or even by means of agent's sense perception.

Note that due to the procedural semantics, our agents can learn new concepts by asking the other agents. In our example, after receiving Adam's reply Berta learns the refined meaning of the 'vacant car park' predicate, i.e., she updates her knowledge base by the respective compound construction defining the property of being a parking lot with vacancies. Moreover, though our approach is as fine-grained as the syntactic approach of languages like KIF, the content of agent's knowledge is not a piece of syntax, but its meaning. And since the respective construction is what synonymous expressions (even of different languages) have in common, agents behave in the same way independently of the language in which their knowledge and ontology is encoded. For instance, if we switch to Czech, the underlying constructions are *identical*: $^0[Vacant^0Car_Park] = ^0[Volné^0Parkoviště]$.

The above outlined method is currently being implemented in the *TIL-Script* programming language, the computational FIPA compliant variant of TIL.⁵ It is a declarative functional language that serves for encoding the content of ontologies, knowledge bases as well as communication of agents. The development of *TIL-Script* language is still a work in progress. The implementation of *TIL-Script inference machine* proceeds in stages. In the first stage we implemented the subset of language corresponding to the expressive power of Horn clauses in order to apply Prolog brain unit. Then we extend it to the full FOL inference machine. The next stage is to implement the inference machine for the subset of classical λ -calculi, and finally, the hyper-intensional features and partiality are to be taken into account. This contribution is thus an optimistic view from the theoretical point of view.

However, from the practical point of view our position is rather realistic than optimistic. There is still a vast amount of problems to be solved and work to be done. Intelligent agents' communication and reasoning cannot be realised without shared ontologies. Yet, as soon as the agents are equipped with the ability to learn, they can come into being with a minimal ontology and functionalities.

4. Conclusion

In the paper we have addressed four core topics under the umbrella of multi-agent knowledge modelling. We discussed the Semantic-Web technologies, reality vs. agents, communication of agents and cross-cultural knowledge modelling. We presented pessimistic, realistic as well as theoretically optimistic view points. In this way we illustrated the complexity and multi-dimensionality of the subject. By way of conclusion we can state that perhaps the most important problems to deal with in order to realize the vision of Semantic Web and multi-agent platform are semantic annotation, shared ontology, communication, fine-grained knowledge representation and learning, and last but not least, realisation of inference machines.

⁵ For details see Ciprich, Duží, Košinár: 'The TIL-Script language'; in this proceedings.

These goals cannot be achieved all of a sudden. Yet we have to aim at step by step improvements of the current situation by way of developing new inference methods, expressive languages and multi-agent techniques.

Acknowledgements. This research has been supported by the grant agency of Czech Academy of Sciences, project No. GACR 401/07/0451 “Semantisation of Pragmatics”, and by the program ‘Information Society’ of the Academy of Sciences of CR, project No. 1ET101940420 “Logic and Artificial Intelligence for multi-agent systems”, and projects No. 1ET100300517, 1ET100300419 and MSM-0021620838.

References

- [1] Adida, B. (2008): Bridging the Clickable and Semantic Webs with RDFa. *ERCIM News* 72, 24-25.
- [2] Allen, J.F. (1983): “Maintaining Knowledge about Temporal Intervals”. *Communications of the ACM*, No. 26, pp. 832-843.
- [3] Annotation at W3C, see <http://www.w3.org/> for Ruby Annotation, GRDDL, RDFa ,...
- [4] Bazire, M. and Brézillon, P. (2005): Understanding Context Before Using It. In: Dey, A., Kokinov, B., Leake, D. and Turner, R. (Eds.), *Modelling and Using Context*. The 5th International and Interdisciplinary Conference, CONTEXT 2005, Paris, France, July 5 – 8, 2005, pp. 29 – 40.
- [5] Bednárek, D., Obdržálek, D., Yaghob, J., Zavoral, F. (2005): Data Integration Using DataPile Structure, In: Proceedings of the 9th East-European Conference on Advances in Databases and Information Systems, *ADBIS 2005*, Tallinn, ISBN 9985-59-545-9, 2005, 178-188.
- [6] Berners-Lee, T. (2008): The Web of Things. *Keynote ERCIM News* 72 (2008) p.3.
- [7] Berners-Lee, T., Hendler, J. and Lassila, O. (2001): The Semantic Web. *Scientific American*, May 2001, pp. 34-43.
- [8] Dedek J. (2008): Extraction of Semantic Information From web Resources, WDS 2008, <http://www.mff.cuni.cz/veda/konference/wds/#PROCEEDINGS>
- [9] Dedek, J., Eckhardt, A., Galambos, L., Vojtas, P. (2008): *Web Semantization*; Technical Report 2008.
- [10] Dey, A. K. (2001): Understanding and Using Context. *Personal and Ubiquitous Computing*, Vol. 5, No. 1., pp. 4 – 7.
- [11] Duží, M. (2004): Concepts, Language and Ontologies (from the logical point of view). In *Information Modelling and Knowledge Bases XV*, Kiyoki, Y., Kangassalo, H., Kawaguchi, E. (eds), IOS Press Amsterdam, 193-209
- [12] Duží, M. (2008): TIL as the Logic of Communication in a Multi-Agent System. *Research in Computing Science*, vol. 33, 27-40.
- [13] Duží, M., Heimburger A. (2006): Web Ontology Languages: Theory and practice, will they ever meet? In *Information Modelling and Knowledge Bases XVII*, Kiyoki Y., Hanno, J., Jaakkola, H., Kangassalo, H. (eds), IOS Press Amsterdam, 20-37
- [14] Duží, M., Materna, P. (2008): Concepts and Ontologies. In *18th European-Japanese Conference on Information Modelling and Knowledge Bases*. Kiyoki, Y. and Tokuda, T. (eds.), Tsukuba, Japan, pp. 45-64.
- [15] Eckhardt A.: Tokaf. <http://sourceforge.net/projects/tokaf>
- [16] Eckhardt A., Horváth T., Maruščík D., Novotný R., Vojtáš P. (2007): Uncertainty Issues in Automating Process Connecting Web and User, in *URSW'07 Uncertainty Reasoning for the Semantic Web*. CEUR-WS.org/Vol-327/paper9.pdf
- [17] Egenhofer, M. J., Rashid, A. and Shari, B.M. (1998): “Metric Details for Natural-Language Spatial Relations”. *ACM Transactions on Information Systems*, Vol.16, No.4, pp.295-321.
- [18] Egothor search engine <http://www.egothor.org/>
- [19] Feigenbaum L., Herman I., Hongsermeier T., Neumann E. and Stephens S. (2007): The Semantic Web. In: Action, *Scientific American*, Nov. 2007, 64-71.
- [20] Galamboš L. (2006): Dynamic Inverted Index Maintenance, in: *International Journal of Computer Science*, Vol. 1, No. 2, International Academy of Sciences, World Enformatika Society, ISSN 1306-4428, pp. 157-162.
- [21] Heimbürger, A. (2008): Temporal Entities in the Context of Cross-Cultural Meetings and Negotiations. In: Kiyoki, Y. and Tokuda, T. (Eds.), 18th European-Japanese Conference on Information Modelling and Knowledge Bases (*EJC2008*), June 2-6, 2008, Tsukuba, Japan, pp. 297 – 315.
- [22] Heimbürger, A. (2008a): When Cultures Meet: Modelling Cross-Cultural Knowledge Spaces. In: Jaakkola, H., Kiyoki, Y. and Tokuda, T. (Eds.). 2008. *Frontiers in Artificial Intelligence and Applications*, Vol. 166, *Information Modelling and Knowledge Bases XIX*. Amsterdam: IOS Press, pp. 314 – 321.

- [23] Hofstede, G. (2003): Geert Hofstede™ Cultural Dimensions (referred 13th Aug. 2007) <URL: <http://www.geert-hofstede.com/>>.
- [24] Hofstede, G. and Hofstede, G. J. (2004): *Cultures and Organizations: Software of the Mind: Intercultural Cooperation and Its Importance for Survival*. New York: McGraw-Hill. 300 p.
- [25] Holden, N. J. (2002): *Cross-Cultural Management. A Knowledge Management Perspective*. Harlow, UK: FT Prentice Hall. 328 p.
- [26] King, W. R. (2007): A Research Agenda for the Relationships between Culture and Knowledge Management. *Knowledge and Process Management*, vol. 14, no. 3, pp. 226 – 236.
- [27] Lewis, R. D. (1999): *When Cultures Collide. Managing Successfully Across Cultures*. London: Nicholas Brealey Publishing. 462 p.
- [28] Materna, P. (1998): *Concepts and Objects*. Acta Philosophica Fennica, Vol. 63, Helsinki.
- [29] Materna, P. (2004): *Conceptual Systems*. Logos Verlag, Berlin.
- [30] Mikulova M., Bemova A., Hajic J., Hajicova E., Havelka J., Kolarova V., Kucova L., Lopatkova M., Pajas P., Panevova J., Razimova M., Sgall P., Stepanek J., Uresova Z., Vesela K. and Zabokrtsky Z. (2006): Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual, Technical Report 30, UFAL MFF UK, Prague, Czech Rep.
- [31] Nečaský M., Pokorný J. (2008): Designing Semantic Web Services using Conceptual Model. *ACM SAC 2008: Proceedings of The 23rd Annual ACM Symposium on Applied Computing*, Volume 3. March 2008. Fortaleza, Ceará, Brazil. pp. 2243-2247.
- [32] Nekvasil M., Svátek V., Labský M. (2008): Transforming Existing Knowledge Models to Information Extraction Ontologies. In Proc. Business Information Systems, *BIS 2008*, W. Abramowicz and D. Fensel eds. Springer series LNBIP 7, 106-117.
- [33] Pokorný, J., Richta, K., Valenta, M. (2008): Cellstore: Educational And Experimental XML-Native DBMS. Chapter in: *The Inter-Networked World: ISD Theory, Practice, and Education*. Barry, C., Lang, M., Wojtkowski, W., Wojtkowski, G., Wrycza, S., & Zupancic, J. (eds), Springer-Verlag: New York, to appear.
- [34] Qiong Luo and Hejun Wu (2007): “System design issues in sensor databases”. *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 1182 – 1185.
- [35] Šváb O., Svátek V., Stuckenschmidt H. (2007): A Study in Empirical and ‘Casuistic’ Analysis of Ontology Mapping Results. In: 4th European Semantic Web Conference (*ESWC-2007*), Innsbruck 2007. Springer LNCS 4519.
- [36] Tichý, P. (1988): *The Foundations of Frege’s Logic*, Berlin, New York: De Gruyter.
- [37] Tichý, P. (2004): *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago.
- [38] Tokuda, T. (2005): A characterization of methods for handling large-scale knowledge resources. In *Symposium on Large-Scale Knowledge Resources (LKR2005)*, pages 9–12, 2005.
- [39] URW3 - Uncertainty Reasoning for the World Wide Web W3C Incubator Group Report 31 March 2008, <http://www.w3.org/2005/Incubator/urw3/XGR-urw3/>
- [40] Wenxing Peng, Taro Terao, Yoshihiro Masuda, Naofumi Yoshida, Jun Miyazaki (2008): “Document History System and its Application to Abnormal Document Access Pattern Detection with a Probabilistic Model”. In proceedings of 22nd International Conference on Advanced Information Networking and Applications (*AINA 2008*), pp. 486-493, Mar. 2008.
- [41] Yoshida N. and Miyazaki J. (2005): “An Automatic and Immediate Metadata Extraction Method by Heterogeneous Sensors for Meeting Video Streams”. IEEE International Symposium on Applications and the Internet (*SAINT 2005*) - the International Workshop on Cyberspace Technologies and Societies (*IWCTS 2005*), pp. 446-449, IEEE Computer Society Press, Feb. 2005.
- [42] Yoshida N., Miyazaki J. (2006): “A Novel Approach to Time-Space-Direction Algebra for Collaborative Work in Ubiquitous Environment”. In proceedings of International Conference on Collaboration Technologies (*CollabTech 2006*), pp. 48-53.
- [43] Yoshida N., Kiyoki Y. and Kitagawa T. (1999): “An Associative Search Method Based on Symbolic Filtering and Semantic Ordering for Database Systems”. *Data Mining and Reverse Engineering: Searching for Semantics*, Part Two, Chapter 6, Spaccapietra Stefano (Editor), Fred Maryanski (Editor), Kluwer Academic Publishers, pp.105-128.
- [44] Yoshida N., Kiyoki Y., Fujishima S., Aiso S. (2005): “An Implementation Method of Knowledge Discovery for SNP and Clinical Databases by Haplotype Analysis”. *DBSJ Letters*, Vol.3, No.4, pp.25-28, March 2005.

Information Modelling and Global Risk Management Systems

Hannu JAAKKOLA^a, Bernhard THALHEIM^b, Yutaka KIDAWARA^c, Koji ZETTSU^c,
Xing CHEN^d and Anneli HEIMBÜRGER^e

^a*Tampere University of Technology (Pori), Finland*

^b*Christian Albrechts University at Kiel, Germany*

^c*National Institute of Information and Communication Technology, Japan*

^d*Kanagawa Institute of Technology, Japan*

^e*University of Jyväskylä, Finland*

Abstract. Utilization of global information resources as a part of risk management is insufficient. The authorities are maintaining information systems mainly for their own purposes, without access to high quality public information sources in Internet and without interoperability between systems of different authorities. Beneficial use of all available information resources would provide an opportunity to create knowledge based on different pieces of information. However, powerful distributed knowledge management, mining of the information items, analysing the quality of them, is needed to create new information to be utilized. The distributed operations needs support of complex network architectures, models supporting mutual understanding over the cultures and language borders, and ability to recognize the context and adapt the results to the new context. This paper opens discussion from different viewpoints to the topic of global risk management. Architectural solutions supporting interoperability, quality of data in wide networks, ubiquity and mobility as well as time dimension of the information space are covered.

1. Introduction

The existing information sources provide a huge amount of information to solve the problems connected to wide catastrophes and disasters. The existing information, however, is

- distributed,
- collected to serve the individual needs,
- provided by different authorities and organisations,
- located sometimes in closed bases.

The information has in many cases also bindings to cultures and contexts, which make the beneficial use of these difficult. Problems are also caused by the low or totally missing interoperability between the systems managing this information, as well as the restrictions for the public use of it.

The existing information infrastructure crosses the geographical borders. Technically the availability of information worldwide is easy and fast. In the same time even the big catastrophes and disasters have common meaning: the consequences are often worldwide, people representing different nationalities are part of it, and the responsibil-

ity to recover the damages is common. Even in the smaller and local accidents the information available from one source would be helpful in solving the problem in another context. Because of that improved information management has high demand in the context of global risk management.

The modes of reactivity in the existing or becoming situation would be classified in the following way. In the *passive mode*, a situation is registered only but it does not cause any actions. This kind of situations may provide information to the existing bases and would be beneficial for later use. A good example is passive earthquake registration to the publicly available databases accessible via Internet. In the *reactive mode*, after registering the situation some (pre-planned) actions are triggered. Most legacy information systems developed to support decision making by authorities are alike. The *preactive mode* provides support for decisions to prepare the responsibilities for the becoming events in advance. Good example of this kind of system is the mud flow warning system introduced later in this paper, or the systems developed for tsunami recognition. In the *proactive mode* the system provides in advance information and guidelines to impact in the becoming event. This needs, in addition to the pre-action, also knowledge how to avoid the becoming event totally or what should be done to decrease the level of damage. The advanced level is the *foresight* – ability to foresee the becoming events in advance. This is usually based on the complex modelling of the situations, integrating information of different sources, complex calculations, and ability to understand the original context of the information and adapt it in new context.

This paper covers some views to support improved levels of reactivity in the connection of global risk management. One of the main elements is the ability to benefit on the information managed by the different legacy systems and public source in a seamless way; this needs architectural solutions based on open interfaces between systems and improved adaptive modelling of data to be provided by one data source to be used by another one, even over cultures and contexts. This topic will be discussed in Chapter 2. The higher understanding of information is based on the advanced knowledge management; this topic is discussed in Chapter 3 by introducing the concept of “Next Generation Web”. Chapter 4 opens the discussion on the role of ubiquity in knowledge processing. In the case of distributed knowledge management there exist a problem of fast availability of knowledge items and ability to connect these in the utilizable form. A Knowledge Grid Platform for Collaborative Knowledge System is introduced in Chapter 5. This Grid-architecture is developed especially to connect to each other local knowledge management systems to the global knowledge grid. As an example in Chapter 6 a proactive risk management system, developed for mud flow warnings is introduced. As time is an essential dimension in our information space and an important resource in the natural disasters. Chapter 7 opens discussion on the Temporal Information Processing in the Context of Global Risk Management.

This paper is based on the presentations of the panel discussion in EJC 2008 Conference.

2. Towards Seamless and Mobile Systems

In the context of global risk management there exists a growing need for collaboration over the borders of cultures. The collaboration is based on the communication in different forms. Because of that we need adaptive and context aware applications, which are widely available in a seamless way. The SSMC/DDKM (Seamless Services and

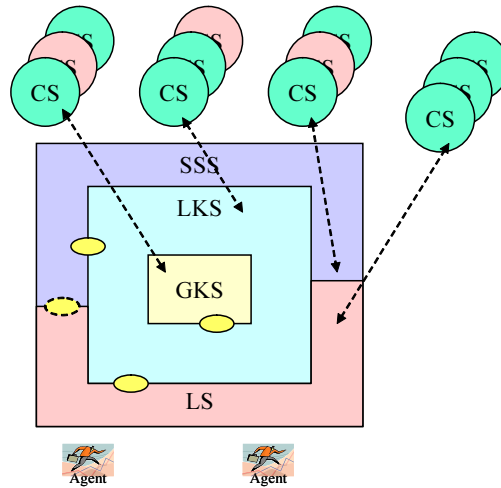


Figure 1. The Three-tier architecture supporting interoperability and distributed knowledge processing.

Mobile Connectivity in Distributed Disaster Management), a project of Tampere University of Technology (Pori), is proposed. The vision of the project is specified: “There exists ways to improve *interoperability* between legacy systems as well as to provide *flexible connectivity* of new services to the integrated whole. Beneficial use of *mobile* devices will have a growing role as instructive, both by push and by demand, devices and as information providers in the case of disasters. Communication will be supported by *models* as a joint language – even over cultural borders. Added value (effectivity, quality of data) is available by higher interoperability and by beneficial use the *global sources* of information (knowledge) in addition to the local ones.”

Interoperability – especially between the information systems of authorities – is typically low. In addition, the ability of the legacy systems to use external information is often either restricted (for safety reasons) or not used in a beneficial way because of missing standardized interfaces. This is true in spite of the fact that several public sources are able to provide such information that would easily be mined and connected to the existing items of information by using improved distributed knowledge processing technologies. Problems are not technical but cultural and organisational. Technically networking of different applications and devices, as well as the usage of variety of terminals in communication is possible. In SSMC project, the concept of “Three Tier Knowledge Management Architecture (as shown in Fig. 1), applying SOA (Service Oriented Architecture) components, is developed. The system components are classified in three categories:

- *General Kernel System (level 0) – GKS*: provides centralized Knowledge Base (KB) and Knowledge Management (KM) services to the next tier nodes and gets processed data from next tier nodes;
- *Local Kernel Systems (level 1) – LKS*: provides local KB and KM services to the next tier nodes, processes data coming from the Sensing SubSystems (SSS) and provides services to the next tier nodes (SSS, LS);
- *Sensing and/or Service SubSystems (level 2.1) – SSS*: sensing = raw data production, service e.g. transportation logistics guidelines; SSS produces (and

processes) data to the upper level systems, gets data from upper level nodes and processes data/provides specialized services;

- *Legacy Systems (level 2.2) – LS*: provides special data and uses upper level services via standardized interfaces;
- *Client Systems (level 3) – CS*: uses services from different levels of nodes based on fixed or service oriented connection to other nodes.
- *Agents - Multi-agent technology* can provide brokering functions and services to for monitoring the whole system and connecting its components to co-operate.

Utilizing the SOA components encourages transfer from applications towards service brokering served by intelligent agents connecting service requests to services available. More detailed the architecture principles are described by Jaakkola in [7].

The concept of *seamless* availability services covers different viewpoints:

- *Seamless = Invisible*: the user does not have to know the services available; in special situations these can be pushed or opened to be available without special demand of the user. This type of services is important especially to guide people in the disaster area to avoid risks and damages.
- *Seamless = Symmetric*: Services are provided by terminal and location independent way. This can be implemented by using standardized interfaces and data channels.
- *Seamless = By demand*: Services are available if needed (service push, information demand).
- *Seamless = Adaptable*: Best service available is utilized instead of the fixed connections.

In practice it is a question on context awareness and adaptive services according to the situation at hand.

The role of *mobile* devices is diverse, because of the fast growing processing capacity and the amount built in properties. Mobile device can be characterized as a sensor, advisor, messenger and its universal usability. The built in properties make it a multi-sensor device: it is location sensitive, able to sensor environmental issues (specialized built in or external sensors connected with RF), record live images (camera) as well as able to make user monitoring (e.g. heart rate recording in co-operation with compatible sensors). Mobile infrastructure is growing towards ubiquity – services are communicating with mobile terminals and able to collect context sensitive data, to make context sensitive knowledge analysis and utilization, also to recognize its user and provide selective messaging. In the ubiquitous world the users have become location independent.

The role of *models* is to support cross-cultural communication as a standardized way to specify the phenomenon (situation). Models for that purpose are semi-formal and universal, able to model behaviour (process models), data, information, knowledge, or structures of them. In disaster management context we speak about situational awareness: it is a description or a specification of a certain situation. Traditionally these are descriptive and based on literal format, which make them language and culture dependent. By providing easy-to-use tools to translate the literal models – or kernel parts of them – in the form of semi-formal models, we are able to win the culture dependency. Cross cultural model based communication is needed in the connection with the wide and large disasters, which are usually global and the recovering activities are in

many cases based on the co-operation of several nations. The *global* dimension of disaster management can also be seen in a way to get beneficial use of the experiences of others. Benchmarking, i.e. experiences moved to new contexts may be useful to solve problems in an innovative way transferred in a new context. Even the information sources are globally available: we need means to merge the pieces of information in an intelligent way and to create new knowledge.

A collection of material handling the topics discussed in this article are available in [8] and in a paper version of the material covering same topics (available in autumn 2008) in the Publication Series of Tampere University of Technology.

3. Intelligent Data Mining and Analysis for Disaster Management

3.1. Conceptual Modelling for Intelligent Data Mining and Analysis

The data mining and analysis task must be enhanced by an explicit treatment of the languages used for concepts and hypotheses, and by an explicit description of knowledge that can be used. The algorithmic solution of the task is based on knowledge on algorithms that are used and on data that are available and that are required for the application of the algorithms. Typically, analysis algorithms are iterative and can run forever. We are interested only in convergent ones and thus need a termination criterion. Therefore, conceptualisation of the data mining and analysis task consists of a detailed description of six main parameters discussed in the following paragraphs.

The data analysis algorithm: A large variety of algorithms has been developed in the past. Each of these algorithms transfers data and some specific parameters of the algorithm to a result. However, algorithms may be restricted in one way or another, e.g., by efficiency or complexity criteria or data quality requirements.

The concept space: The concept space defines the concepts under consideration for analysis. These concepts are modelled within a certain language and can be characterised by certain criteria. Analysis typically target on those criteria that can be either supported or rejected by the data. Furthermore, concepts may be underspecified and be a target of analysis. Concepts may be refined in a variety of ways, e.g. inductively, by generalisation and classification, by instantiation and by contextualisation.

The data space: The data space typically consists in a multi-layered data set of different granularity. Data sets are typically only small samples compared with all the data that should be considered. We need therefore to know which generalisation, extrapolation and abstraction techniques can be applied to the data. Also the quality of the samples and the unknown (or known) probability distribution of values must be considered. The data space is often describable through certain database schemata. Data sets can be chosen systematically or could be chosen maliciously. Data sets may be enhanced by metadata that characterise the data sets and associate the data sets to other data sets. The data space allows applying some data exploration techniques such as roll-up or dice operations, querying techniques such as tree queries for separation. Some data sets allow getting information concerning the concept by actively experimenting with it. A very important issue is whether or not the analysis model can handle noisy or erroneous data sources.

The hypothesis space: Generally, an algorithm is supposed to map evidence on the concepts to be supported or rejected into a hypotheses about it. Therefore, one has to choose a set of possible descriptions. Clearly, each criterion contained in the concept

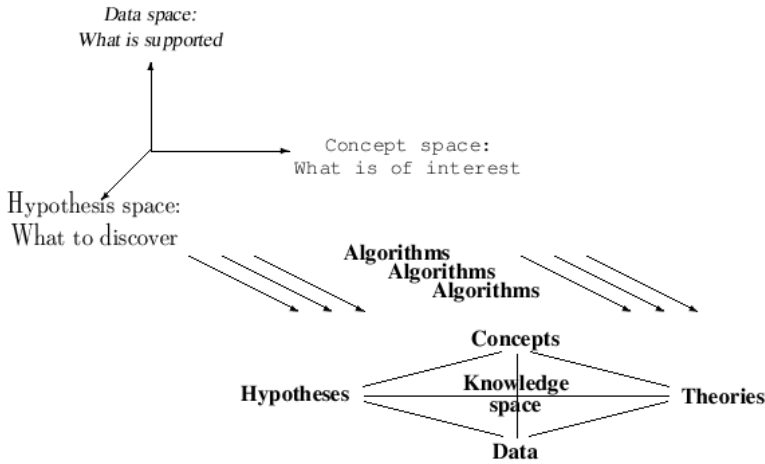


Figure 2. The Knowledge Detection Process of Data Mining and Analysis.

space has to possess at least one description in the hypothesis space. However, the hypothesis space may additionally contain descriptions not describing any concept in the concept space. Furthermore, the descriptions provided by the hypothesis space may be slightly different from those ones used in defining the concepts in the concept space.

The prior knowledge space: Here, one has to specify which initial knowledge about the domain the algorithm may use. This generally restricts the analysis uncertainty and/or biases and expectations about the concepts to be analysed. Obviously, specifying the hypothesis space already provides some prior knowledge. In particular, the analysis task starts with the assumption that the target concept is representable in a certain way. Furthermore, prior knowledge may also be provided by “telling” the algorithm that “simple” answers are preferable to more “complex” hypotheses. Finally, looking at important applications one has to take into account that prior knowledge may be “incorrect.” Thus, when developing advanced analysis techniques one has to deal with the problem how to combine or trade-off prior versus new data sets.

The success criteria: Finally, one has to specify the criteria for successful analysis. This part of the specification must cover at least some aspects of our intuitive understanding of analysis. In particular, we have to deal with questions like: “How do we know whether, or how well, the analysis was successful?” “How does the algorithm demonstrate that hypotheses are supported by the concepts and the data?”

Each instantiation and refinement of the six parameters described above leads to specific data mining task. The result of data mining and data analysis is described within the knowledge space. The data mining and analysis task may thus be considered to be a transformation of data sets, concept sets and hypothesis sets into chunks of knowledge through the application of algorithms. We visualize this process in Fig. 2.

3.2. The Kiel Data Mining Workbench

3.2.1. Towards Quality-Driven Data Mining and Analysis

Intelligent data mining and analysis consists of provision of data at an adequate level of detail and an adequate level of quality and on application of techniques for mining and analysis of data, content, information or knowledge. Knowledge explication addition-

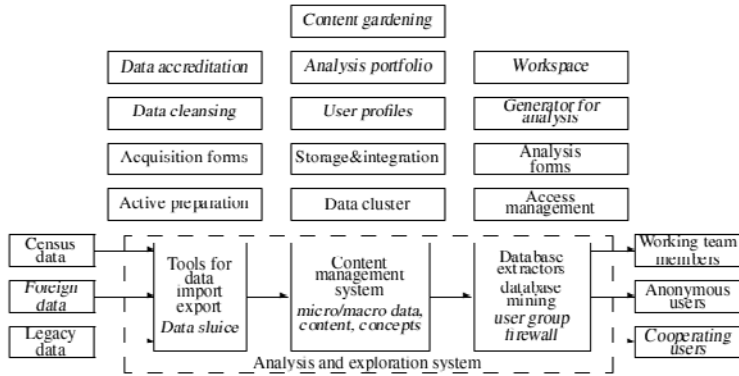


Figure 3. The Intelligent Data Gathering, Storage, and Analysis Workbench.

ally requires validation, verification and explanation of the concepts obtained. Data are at an adequate level of details if they are either abstracted in such a way that they become analysable and are at an appropriate level of quality for the application of algorithms or are concretised in such a way that the phenomena may be discovered. We therefore need a facility to dive into micro- or meso-data or to roll up to macro-data. Macro-data may again be considered to be micro-data within a suite of models. Micro data or raw data or sensor data are analysed according to the quality criteria of data analysis algorithms. Macro-data typically also serve as a facilitating means for explanation of results. Quality-driven data mining and analysis therefore consists of a number of typical interrelated with each other tasks discussed below.

Gathering, gardening of content depends on level of data (macro), meta-data, validity, timeliness, recharging, versions, and quality. We may use approaches that have been developed for data warehouse technology, e.g., for play-in and play-out of data. Algorithm is a field of research in computer science that develops patterns of algorithms and clarifies their application to the derivation of specific algorithms. Data mining has resulted in '1001' data mining algorithms. These algorithms may be classified in dependence of quality, of data and of their profile and their portfolio.

Meaningful and reasonable interpretation is necessary before developing the concept space and the hypothesis space and after obtaining analysis results. Otherwise, we lack in explanation facilities of the data mining and analysis results.

Strategies can be developed for further elaboration of data mining and analysis depending on the results that have been obtained so far.

3.2.2. Tool support for Intelligent Data Mining and Analysis

Intelligent data mining and analysis is a complex task that must handle all three aspects of data management: input, storage, and export. We are therefore currently developing a workbench that extends the classical data warehouse architecture by certain tools. Data warehouses use a rigid separation of the three aspects and architecture with an input, storage and export machine. The architecture is displayed in Fig. 3.

The workbench consists of a number of specialised tools such as the following. *The data analysis workbench* provides an intelligent support for data analysis, concept development and hypothesis proliferation. It follows the approach depicted in Fig. 3.

We are interested in reliable and defeatable analysis results. *The data import and improvement facilities* are based on generic data import forms, support detection of data massive that can be integrated with existing data and are going to provide automatic importers for foreign, census or legal data. *The data export and collaboration facilities* are based on query forms similarly to those developed for ER-based data processing and annotate data with additional metadata such as a citation track. The data warehouse architecture for informed users with survey on available data massive and their usage (conditions, facilities ...); export interfaces and tools. *Effective access control* – role-, portfolio-, profile- and collaboration-based rights and obligations, and protection against anybody else combined with world-wide exclusive use for partners with citation of usage. *Intelligent integration* of foreign, legacy and new data includes transformer for foreign and legacy data, and data gardener for consistent data protection and up growth

4. Knowledge Processing in Ubiquitous Computing Environment

Ubiquitous networks provide network universal connectivity. Users can acquire information in the form of digital content anywhere, anytime. Currently, even CPUs on mobile devices have sufficient power to process sound and motion picture data. Multimedia content is already being used widely over various networks.

Nevertheless, the text information, such as web content, blogs, and email, is standard, and most people rely on this information. Consumer-generated media (CGM) provide a wider variety of text with multimedia content. Such information can be used by the general users in their daily lives, and is slowly casting a strong influence on government policies and enterprise management.

In the ubiquitous computing environment, we publish information obtained from ubiquitous devices in the real-world on the Internet. Various network services collect user-generated information. These services sense the information, analyze it, extract knowledge, correlate information (or correlate information with a physical object), create digital contents for each user, and display the content on ubiquitous devices. Users can publish and distribute more information when they browse the generated content on these devices. This circulation of digital content should be controlled by the user's request in real-time, that is, distribution of seamless content in the next-generation Web.

Ambient intelligence is one of the novel information processing technologies. The technology will become invisible, embedded in our natural surroundings, presents whenever we need it. We will operate this by simple and effortless interactions. This information will be attuned to all our senses, adaptive to users, and context-sensitive over multiple devices. Ambient intelligence will be the core technology for Web 3.0. These approaches help in realizing Ambient Intelligence. A complete overview of the same is shown in Fig. 4.

In order to realize the above content operation, we must analyze, search and create digital content according to the user's context. Since we often decide our subsequent actions on the basis of the information obtained in the real world, this information should be credible and useful. Furthermore, with the aid of seamless operations, we must enhance the quality of the obtained information.

We have already developed a functional web architecture, which enables us to acquire digital content from natural surroundings, facilitate seamless searching of more

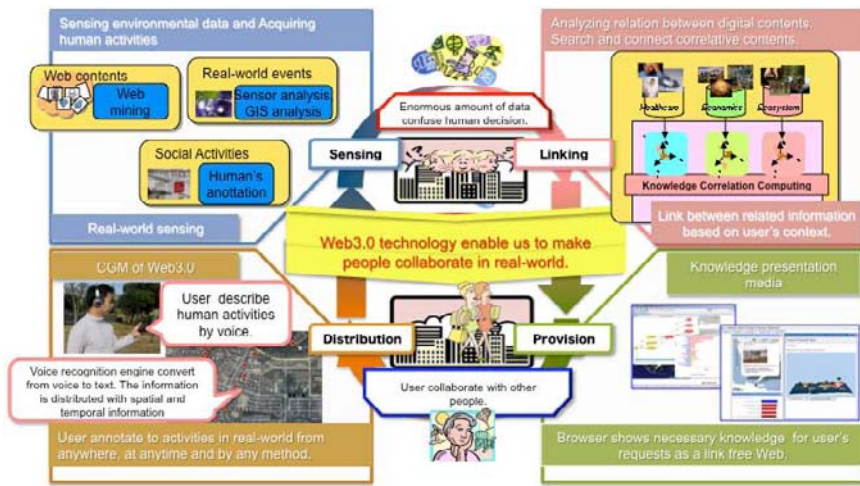


Figure 4. Circulation of Knowledge Processing for Ambient Intelligence.

related information on several ubiquitous devices, and edit CGM content automatically and publish information to the Internet [10,11].

5. Knowledge Grid Platform for Collaborative Knowledge Systems

In emergency management systems, past and future objectives remain the same: “providing relevant communities collaborative knowledge systems to exchange information”. Various communities organize their own knowledge repositories, each of which aggregates perception, skills, training, common sense, and experience of a community of people [21]. Thus, knowledge sharing, searching, analysis and provision are essentially important for realizing knowledge-based modern societies with various knowledge processing facilities in a world of networks. Social and policy issues are to be addressed with the idea of coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. In “Knowledge Cluster Systems” project at National Institute of Information and Communications Technology (NICT), “3-Sites model for Long-Distance Knowledge Sharing, Searching, Analysis and Provision System” is proposed as a core system model consisting of three essential functions distributed in a global area network environment [20] – see Fig. 5. In this system model, three functional sites are dynamically connected as event-sensing, knowledge analysis, and knowledge provision, respectively, and those sites transmit significant knowledge related to accidental or irregular events from various knowledge resources to actual users. The important feature of this model is to dynamically connect event-sensing, knowledge analysis, and knowledge provision sites, according to occasional contexts in various areas related to accidental or irregular events occurred in global, social and natural environments.

Open-access and easy adaptability of emergency management systems will play an increasingly important role. The “global knowledge grid” is an integrated infrastructure of the knowledge cluster systems for coordinating knowledge sharing and problem

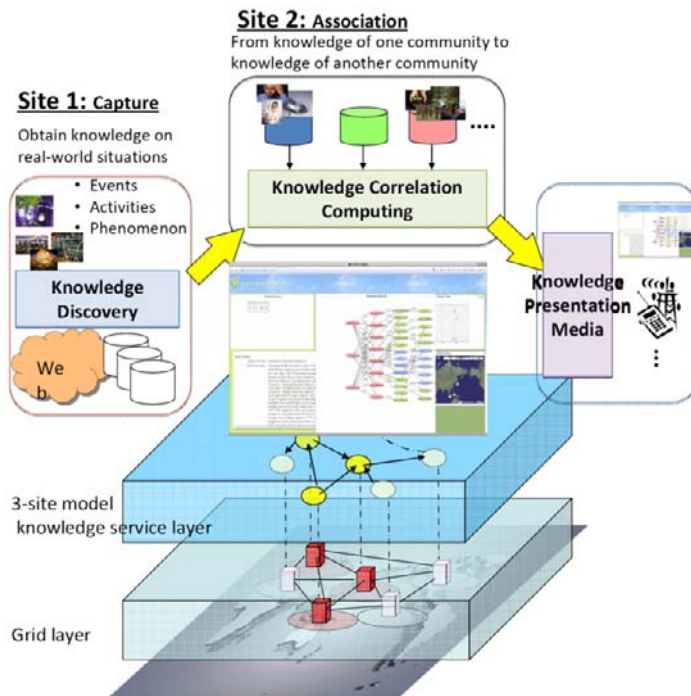


Figure 5. Global knowledge grid model for 3-site knowledge cluster system.

solving in distributed environments [22,23]. The knowledge grid uses the basic functions of a grid and defines a set of additional layers to implement the functions of distributed knowledge discovery, analysis and provision. Those functions are implemented as software modules on the grid nodes in parallel on the basis of a service-oriented architecture (SOA) [16]. The knowledge grid enables collaboration between knowledge providers who must mine data stored in different information sources, and knowledge users who must use a knowledge management system operating on several knowledge bases. Virtual organizations (VOs) form logical groups of the knowledge grid services, in each of which disparate organizations or individuals collaborate in a controlled fashion to achieve a common goal.

The reach of the Internet expands opportunities for public involvement, regardless of physical distance from the disaster area. In the same way as the World Wide Web, the knowledge grid provides a framework of infinitely-evolving knowledge repository by connecting heterogeneous knowledge bases owned by different organizations and communities [15]. A typical example of the connection is based on causal relation from one knowledge base to another knowledge base. For instance, a disaster knowledge base can be connected with healthcare knowledge base by establishing a causal relation in order to find diseases caused by specific disasters. In that way, “a web of knowledge” will be formed as collaboration architectures on demand with the collective intelligence, where collaborative data are treated with social interaction and community management. That is what we call the next generation of World Wide Web or “Web 3.0”.

6. A Mudflow Warning System

6.1. Overview

The mudflow is a dangerous and harmful disaster. It engulfs villages, factories, railways and roads. An important research issue is how to establish a mudflow warning system for sending messages to the people in the dangerous areas when the disaster occurred. In this paper we represent our idea for establishing a mudflow warning system. In the system, monitor cameras are used as sensors for catching the signals of the disaster. The idea is also represented on how to recognize the vision signals of the mudflow and how to broadcast the warning message

In May 2006, a mudflow engulfed villages, factories, railway and roads in an area nearby Surabaya, the second city of Indonesia. Many people have been driven from their homes by a torrent of hot toxic mud. The people who are late for escaping from the mud torrent lost their precious lives. It is an important issue to establish a mudflow warning system for reducing the damage of the disaster. It is also an important research issue on how to find the mudflow as soon as possible when it flows out from the mouth of the mud volcano and how to broadcast a warning message immediately to the people, who need the information for escaping from the torrent of the hot toxic mud.

When the hot mud flows out from the mouth of the mud volcano, hot stream gas also blows out from the mouth. As the vision signal of the hot stream of the mudflow can be caught by cameras far from the volcano mouth, it is possible to establish a warning system by using monitor cameras as the sensors of the system. When the signals of a mudflow happened far from the monitor cameras are caught and the warning messages are broadcasted immediately, people will be given enough time escaping from the dangerous areas. Another advantage of using the monitor cameras is that the monitor cameras can be used to monitor the flowing of the mud. In this paper, a mudflow warning system with monitor cameras as sensors is proposed. The outline of the system and the technique for recognizing the vision signal of the mudflow are represented in Section 6.2 and 6.3.

6.2. The Mudflow Warning System

An experimental system with monitor cameras, a vision signal analyzer and a warning message sender is developed. As shown in Fig. 6, in the system, image signals are stored into a video database and transmitted to the vision signal analyzer which analyzes and recognizes the mudflow vision signals.

Positions of the monitoring cameras, e-mail addresses of computers and the positions of the computers are registered in the database of the warning message sender. By using the position information of the monitoring cameras, the position of the volcano mouth of the mudflow can be determined. Based on the position information of computers, the e-mail addresses for broadcasting the warning message can be determined. In our system, cell-phones with GPS are also registered for broadcasting the warning messages to those cell-phones in the dangerous areas.

6.3. Basic Idea for Recognizing Mudflow Image Signal

In order to recognize the mudflow vision signals, vision features, color and image's edges and their position information, are derived automatically from the monitoring

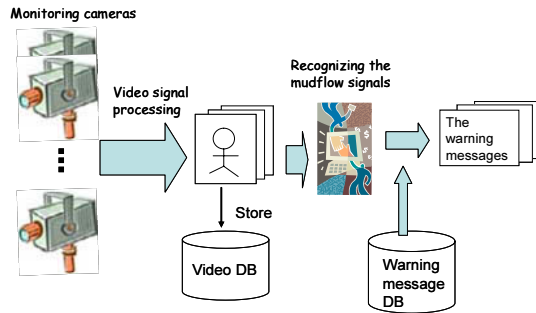


Figure 6. The Mudflow Warning System.

images in the vision signal analyzer. Independent factors of the vision features are extracted by using a mathematical method referred to as the Singular Value Decomposition. 300 images are used to determine the factors correlated to the mudflow vision signals. Based on our experimental results, the factors correlated to the mudflow vision signals are successfully extracted. More details on the feature deriving and the factor extracting are described in [3].

After the mudflow correlated factors \mathbf{F} are extracted, a vector space is constructed by the factor. Monitoring images are mapped onto the vector space by using the expanding calculation $\mathbf{M} * \mathbf{F}$, where ' \mathbf{M} ' is an image vector and '*' is the vector multiple. Normal of each image vector on the vector space is calculated. When the mudflow vision signals are contained in an image, the normal of the image vector will be greater than a threshold. This characteristic is used for recognizing the mudflow vision signals.

7. Time Dimension of the Information Space

7.1. Temporal Information Processing in the Context of Global Risk Management

Time is an essential dimension of our information space. Temporal information processing (TIPS) has an important role in designing and implementing global risk management applications. In the context of global risk management, temporality can have long and medium term dimensions such as identification of causal relations between disasters and certain diseases. Temporality can also have very time-sensitive, short-term dimensions, for example sharing and updating information about rescue operations. Temporal information processing in the context of global risk management can be studied on four levels: general level, content level, functional level and system level [4].

General level: When modelling time, there are two main traditions represented in the literature. One view of time is a set of points without duration. The other is that intervals should be considered as temporal individuals. There are some general time ontologies such as OWL-Time. OWL-Time is ontology of temporal concepts. The ontology provides a vocabulary for expressing facts about topological relations among instants and intervals, together with information about durations, date times and time zones. OWL-Time has been extended to cover temporal aggregates as well. Temporal aggregates are collections of temporal entities. Perdurants, on the other hand, are entities that are only partially present, in the sense that some of their proper temporal parts

(e.g., their previous or future phases) may be not present. Perdurants are often known as processes, for example a “rescue operation”. The essential concepts of OWL-Time, temporal aggregates and perdurant ontology are summarized in [5]. Time ontological approaches in the GRM context can be applied to formalize temporal contents of Web resources and to describe temporal properties and functions of Web services.

Content level: Natural language-based information systems and knowledge management, which can take advantage of temporal dimensions of information and knowledge, can perform many useful functions. Applications such as temporal information recognition and extraction, question-answering, summarization and visualization can all benefit from analysis and interpretation along the temporal dimensions. In such applications, information and knowledge should be transformed into temporally aware structures that can then be used to solve application-related problems. Temporal markup languages are used to transform pieces of information and knowledge into temporally-aware structures [19]. For example TimeML (Markup Language for Temporal and Event Expressions) is a robust specification language for events and temporal expressions in natural language. Other interesting methods for processing temporal semantics of pieces of knowledge are: (a) Allen’s relations between time intervals, which can be applied to calculate temporal relations between pieces of knowledge [1], (b) topic detection and tracking of news materials, which can be used for indentifying causal relations between temporal phenomena [14] and (c) temporal data mining that concerns with large sequential data sets for example time series [13]. The knowledge content of a global risk management application requires efficient functions that also include temporal awareness for supporting time-sensitive knowledge sharing, analysis and delivery among remote sites.

Functional level: Modelling temporal variations of data and temporal pattern recognition are important issues in global risk management applications. Snapshots databases only contain current data, which are a snapshot of the current reality. Many application, however need both current and past data and possibly future as well. In the broadest sense a database that maintains past, present and future data is called a temporal database. The activities of rescue organizations, for example the management of rescue operations, are ongoing processes and their information needs and processing capabilities should be considered in a time perspective. That is, to support managerial information needs, as well as others, the relevant knowledge bases should possess a temporal dimension to store, analyze, share and deliver time-varying data. Most data models however do not address issues of maintenance and processing of temporal data. These models either create undue data redundancy and/or provide limited time-processing capacity. These are two possible directions that can be followed for handling temporal data. One is to develop a new model to support time dimension and the other to augment existing data models to support time dimension in a coherent way [9,18]. Global risk management requires efficient database models and functions that also include temporal awareness for supporting knowledge sharing, analysis and delivery among remote sites.

System level: A Petri net is a formal method for modeling functions and information flows in distributed systems [17]. As a modeling language, it graphically depicts the structure of a distributed system as a directed bipartite graph with annotations. As such, a Petri net has place nodes, transition nodes, and directed arcs connecting places with transitions. The places from which an arc runs to a transition are called the input places of the transition. The places to which arcs run from a transition are called the output places of the transition. In certain cases, the need arises to also model the timing,

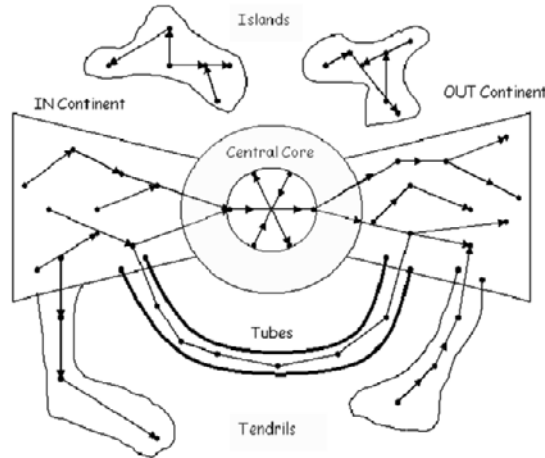


Figure 7. Web continents [2].

not only the structure of a model. For these cases, timed Petri nets have evolved, where there are transitions that are timed, and possibly transitions which are not timed. In the context of global risk management, Petri nets and timed Petri nets provide an interesting approach to model a large distributed system and its time-sensitive parts.

To summarize aspects of temporal information processing in the GRM context, we can present three relevant questions for further research: (1) What kind of temporal and time-sensitive knowledge structures can be identified in the GRM-context? (2) What kind of time-sensitive functions and services are needed in the GRM-context? (3) What kind of models and methods we have for temporal information processing in the GRM-context?

7.2. Web Continents and Levels of Linking in the Context of Global Risk Management

The World Wide Web does not form a single homogeneous network. Rather, according to [2], it is fragmented and broken into four major continents (Fig. 7). Each continent has traffic rules of its own when we want to navigate Web lands. In Central Core each node can be reached from every other node. The nodes of IN Continent are arranged such that following the links eventually brings the user back to Central Core. However, when the user starts from the Core he/she is not allowed to return to the IN Continent. In OUT Continent, all nodes can be reached from the Core. Once the user has arrived OUT, there are no links taking her/him back to the Core. Tubes can connect the IN and OUT Continents. In Tendrils, some nodes attach only to IN and OUT Continents. Nodes of Isolated Islands can not be accessed from the rest of the nodes. They are isolated groups of interlinked resources that are unreachable from the Central Core and do not have links to it.

These four continents significantly limit the Web's navigability. For example, starting from a node belonging to the Central Core, we can reach all resources belonging to this major continent. IN land and isolated islands cannot be reached from the Core. Is this fragmented structure here to stay? Will the future Web eventually integrate the four continents into one? The answer is simple: As long as the links remain directed, such homogenization will never occur. Search engines can not function effectively.

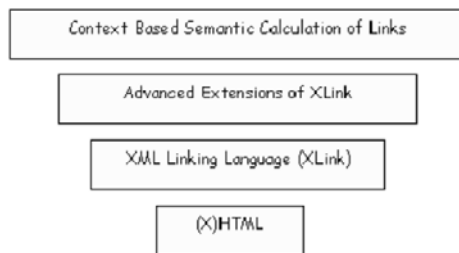


Figure 8. Level of linking in the future Web.

Advanced linking languages such XML/XLink and its extensions [6] can help us in closed Web applications, such as extranets and intranets, which can be parts of global risk management applications. The situation will change radically when we move from closed applications to open Web environments. XLink type languages, even semantically rich, are not anymore enough. We need to extend the approach. We need context based, semantic calculation of links and knowledge in addition to stable, backbone like, knowledge structures [12]. As shown in Fig. 8, we can identify four main levels of linking in the future Web: (a) simple (X)HTML based linking, (b) XML Linking Language (XLink) based level with some link semantics, (c) advanced extensions of XLink with richer link semantics and functionality and finally, (d) context based semantic calculation of links.

8. Summary

The paper is produced in collaboration with the authors, all of them participating the collaborative research projects under the umbrella of “global risk management”. The items handled cover the viewpoints to the solutions supporting seamless interoperability and availability of services, mobile dimension as a part of the information infrastructure, combined ubiquity and mobility, GRID architecture supporting the complex and distributed knowledge processing, and quality of data mined from different sources., In addition the role of time dimension as a part of the information infrastructure was discussed. A concrete proactive risk management application – mud flow control and warning system was introduced. It combines computer vision and knowledge processing in the real disaster situation.

One of the consequences binding the parts of the paper together is the concept of context and the role of modeling. Modelling is seen to be a common language between systems and people – even over the cultures and language borders. However, understanding the context of the information is needed to be able to analyze it in a right way. In addition, ability to adapt the results of analysis and utilize the information in a new context is needed. In conceptual modeling terms “*active conceptual modeling*” and “*context aware adaptive systems*” have become important research topics. Simple Google search provides the following results: “Active Conceptual Modelling” 1,090,000 Google fits, “Context aware adaptive system” 1,760,000 Google fits (in June 2008). Conceptual models are not fixed but adaptive based on learning the phenomena of the environment they are used in. E.g. in the connection with counter-terrorism it is important to notify, based on the exiting models, the risks approaching. A kind of reference model is used as a prescriptive model to provide opportunity to react in the de-

tected risks. In the case the shape of the risk is changing, the model has to learn about the changes and adapt in the new situation – otherwise it is useless.

The joint activity of the researcher (authors of this paper) is continuing to study and find new technologies, methods and approaches to be applied in connection with the distributed risk management. Main items are modelling technologies, intelligent next generation web environment including ability to mine good quality data taking into account the time dimension, grid architectures supporting complex distributed knowledge processing, and software architectures supporting flexible service based connectivity between systems and services.

References

- [1] Allen, J.F. Time and Time Again: The Many Ways to Represent Time. *International Journal of Intelligent Systems*, 6, 4 (1991), 341–355.
- [2] Barabasi, A.-L., *Linked, The New Science of Networks*. Perseus Publishing: Cambridge, MA, USA, 2002.
- [3] Chen, X. and Delvecchio, T., et al., Deriving Semantic from Images Based on the Edge Information, *Information Modelling and Knowledge Bases XVII* (IOS Press.), Vol. 136, 2006, 260–267.
- [4] Heimbürger, A., Temporal Information Processing in the Context of Knowledge Cluster Systems. In: Jaakkola, H. (eds.) *Proceedings of the 2nd International Workshop on Knowledge Cluster Systems – Design for Knowledge Sharing, Analysis and Delivery among Remote Sites*, March 17th – 19th, 2008, Pori, Finland, 14 p. (to appear). 2008.
- [5] Heimbürger, A., Temporal Entities in the Context of Cross-Cultural Meetings and Negotiations. In: Kiyoki, Y. and Tokuda, T. (Eds.) *Proceedings of the 18th European – Japanese Conference on Information Modelling and Knowledge Bases*, June 2 – 6, 2008, Tsukuba, Japan. 2008, 297–315.
- [6] Heimbürger, A. et al., Time Contexts in Document-Driven Projects on the Web: From Time-Sensitive Links towards an Ontology of Time. In: Duzi, M., Jaakkola, H., Kiyoki, Y. and Kangassalo, H. (eds.). *Frontiers in Artificial Intelligence and Applications*, Vol. 154, *Information Modelling and Knowledge Bases XV*. Amsterdam: IOS Press. 2007, 136–153.
- [7] Jaakkola, H., Software Architectures and the Architecture of Long-Distance Knowledge Sharing, Analysis and Delivery Platform. *Proceedings of the First International International Symposium on Universal Communication (ISUC)*, Kyoto, Japan, June 14–15, 2007 (6 pages). Invited paper.
- [8] Jaakkola, H., Soini, J. and Leppäniemi J., Service, Sensor and Mobile Connectivity in Distributed Disaster Knowledge Management (SSMC/DDKM). In Jaakkola H. (ed.), *Proceedings of the Second International Workshop on World-Wide Knowledge Sharing and Analysis – KC2008*. CD-publication. TTY Pori, Publication 9. 2008. ISBN 978-952-15-1948-2. ISSN 1795-2166. 2008.
- [9] Jensen, C.S. et al., The Consensus Glossary of Temporal Database Concepts. In: Etzion, O., Jajodia, S. and Sripada, S. (Eds.): *Temporal Databases – Research and Practice*. LNCS 1399. 1998, 367–405. Springer-Verlag: Berlin Heidelberg.
- [10] Kidawara, Y., Uchiyama, C.T. and Tanaka, C.K., An Environment for Collaborative Content Acquisition and Editing b,TM Coordinated Ubiquitous Devices. *Proceedings of the 14th International World Wide Web Conference (WWW2005)*. 2005, 782–791.
- [11] Kidawara, Y. and Tanaka, K., Cooperative Device Browsing through Portable Private Area Network, *Proc. of the 7th International Conference on Mobile Data Management (MDM2006)*. 2006.
- [12] Kiyoki, Y. and Kawamoto, M., Semantic Associative Search and Space Integration Methods Applied to Semantic Metrics for Multiple Medical Fields. In: Duzi, M., Jaakkola, H., Kiyoki, Y. and Kangassalo, H. (eds.). *Frontiers in Artificial Intelligence and Applications*, Vol. 154, *Information Modelling and Knowledge Bases XV*. IOS Press, Amsterdam, 2007, 120–135.
- [13] Laxman, S. and Sastry, P.S., A Survey of Temporal Data Mining. *Sādhanā*, Vol. 31, Part 2, 2006, 173–198.
- [14] Mori, M., Miura, T. and Shioya, I., Topic Detection and Tracking for News Web Pages. In the *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. 2006.
- [15] Nakanishi, T., Zettsu, K., Kidawara, Y. and Kiyoki Y., Towards Interconnective Knowledge Sharing and Provision for Disaster Information Systems –Approaching to Sidoarjo Mudflow Disaster in Indonesia–, *Proceedings of the 3rd Information and Communication Technology Seminar (ICTS2007)*, Surabaya, Indonesia, 2007, 332–339.

- [16] Papazoglou, M.P. and Georgakopoulos, D., Service-Oriented Computing, *Communications of the ACM* **46**, 10 (2003), 24–28.
- [17] Petri, C.A., *Kommunikation mit Automaten*. Ph. D. Thesis. University of Bonn. 1962.
- [18] Snodgrass, R. and Ahn, I., A Taxonomy of Time in Databases. In: *Proceedings of the 1985 ACM SIGMOD International Conference on management of Data*, Austin, Texas, United States, 1985, 236–246.
- [19] TimeML. TimeML – Markup Language for Temporal and Event Expressions (referred June 11th, 2008) <http://www.timeml.org/site/index.html/>. 2008.
- [20] Zettsu, K., Nakanishi, T. Iwazume, M., Kidawara, Y. and Kiyoki, Y., Knowledge Cluster Systems for Knowledge Sharing, Analysis and Delivery among Remote Sites, *Information Modeling and Knowledge Bases Vol. XIX*, IOS Press, 2008, 282–289.
- [21] Zettsu, K. and Kiyoki, Y., Towards Knowledge Management based on Harnessing Collective Intelligence on the Web, *Proceedings of the 15th International Conference of Knowledge Engineering and Knowledge Management – Managing Knowledge in a World of Networks – (EKAW2006)*, Lecture Notes in Computer Science Vol. 4248. 2006, 350–57.
- [22] Zettsu, K., Nakanishi, T., Iwazume, M., Kidawara, Y. and Kiyoki, Y., Global Knowledge Grid: An Infrastructure for Knowledge Sharing and Analysis – Towards Knowledge Management based on Harnessing Collective Intelligence –, *Proceedings of the 1st International Symposium on Universal Communication*, Kyoto, Japan, 2007, 140–143.
- [23] Zhang, R., Zettsu, K., Kidawara, Y. and Kiyoki, Y., SIKA, A Decentralized Architecture for Knowledge Grid Resource Management, *Proceedings of International Workshop on Information-explosion and Next Generation Search (INGS2008)*, Shenyang, China 2008.

This page intentionally left blank

Subject Index

algorithms	224	inheritance	354
analytical vs. empirical concept	45	intelligence	392
analytical vs. nomic necessity	45	intension	45
annotation	411	intersubjectivity	348
argumentation games	309	IS	331
autonomous agents	411	IS-A hierarchies	65
BFO/Span Ontology	290	ITM	331
blog	212	kansei	384
board games	309	knowledge modelling	411
brain architecture	261	knowledge service	224
business network	366	language model	392
communication	166, 348, 411	language origination	392
concept	45	logical analysis	166
concept formation	392	logical analysis of natural language	45, 411
concept modeling	366	Lyee theory	359
conceptual modeling	65	mathematical model of meaning	384
conjoint analysis	379	Mentalese	373
constraints	85	metalogue	309
co-reference	224	model management	354
cross-cultural knowledge	411	modeling	85
cross-cultural meetings and negotiations	290	monadistic view	245
CSIS	290	mouth articulation	373
culture-sensitive information systems	290	multi-agent system	166, 261
datalog	354	museum	379
description logic	245	musical instrument	379
end user development	359	neural network	392
entity/relationship model	245	news article extraction	194
expectancy	384	news article page collection	194
expertise location	212	news directory	194
explicit knowledge	321	news index system	194
extension	45	ontology	45, 366, 411
film	384	opera	384
focal awareness	321	overlay clustering	113
formal ontologies	245	OWL-Time	290
game trees	309	Peer-to-Peer systems	113
Gunji's systematics	379	perdudants	290
hyper-intension	45	<i>phonosemes</i>	373
index word list construction	194	polymorphism	354
information extraction	180	preference tendency	379
information security	366	prolog	261
		propositional logic	85

reasoning	261	temporal aggregates	290
relationships and properties	245	temporal entities	290
routine knowledge	348	temporal regions of time	290
semantic approach	331	TIL	45
semantic web	411	TIL-Script language	166, 261
semantics	85	transparent intensional logic	166, 261
shared cognition	348	UML	331
similar peers	113	universality	373
sounds meaning	373	web application	180
spatial analogy	373	web service	180
subsidiary awareness	321	XML	65, 85
tacit knowledge	321	XML schema	65
tagging	212		

Author Index

Aaltonen, J.	366	Lai, W.S.	212
Atzeni, P.	354	Li, H.	113
Brezillon, J.	154	Link, S.	85
Brezillon, P.	154	Liu, B.	194
Buřita, L.	331	Mäkinen, T.	123
Chen, X.	v, 105, 429	Materna, P.	45
Chou, S.-C.	212	Medvedev, A.	373
Ciprich, N.	166, 261	Meghini, C.	224
Doerr, M.	224	Mustonen, P.	366
Dror, I.E.	340	Nakagawa, Y.	194
Dubnov, S.	384	Necasky, M.	65
Duží, M.	45, 166, 261, 411	Noro, T.	194
Engelbrecht, P.C.	340	Ohsuga, S.	392
Eriksson Lundström, J.	309	Ondryhal, V.	331
Fiedler, G.	1, 123	Pokorny, J.	65
Fischer Nilsson, J.	245, 309	Sasaki, J.	359
Frydrych, T.	261	Sasaki, S.	105
Funyu, Y.	359	Shao, X.	113
Gianforme, G.	354	Shih, C.C.	212
Hachour, H.	348	Shirota, Y.	379
Hamfelt, A.	309	Spyratos, N.	224
Han, H.	180, 194	Stu, J.	212
Hartmann, S.	85	Takahashi, Y.	270
Hausser, R.	22	Tanaka, M.	359
Heimbürger, A.	290, 411, 429	Thalheim, B.	1, 123, 429
Hsieh, W.-T.	212	Tijus, C.	154
Ibradzic, S.	113	Tokuda, T.	v, 113, 180, 194, 411
Itabashi, Y.	105	Trinh, T.	85
Jaakkola, H.	v, 123, 429	Tsai, T.-M.	212
Kawamoto, M.	253	Vaneková, V.	139
Kidawara, Y.	429	Varkoi, T.	123
Kiyoki, Y.	v, 105, 253, 270, 384	Virtanen, I.	321
Kohut, O.	261	Vojtáš, P.	139, 411
Košinár, M.	166, 261	Yamada, K.	359
Kotake, Y.	180	Yoshida, N.	v, 411
Krone, O.	366	Zettsu, K.	429

This page intentionally left blank

This page intentionally left blank

This page intentionally left blank